IJNS

International Journal
of Network Security

ISSN 1816-353X (Print)
ISSN 1816-3548 (Online)

Vol. 20, No. 2 (Mar. 2018)

INTERNATIONAL JOURNAL OF NETWORK SECURITY

PUBLISHING OFFICE

Min-Shiang Hwang
Department of Computer Science & Information Engineering, Asia University, Taichung 41354, Taiwan, R.O.C.
Email: mshwang@asia.edu.tw

International Journal of Network Security is published both in traditional paper form (ISSN 1816-353X) and in Internet (ISSN 1816-3548) at http://ijns.jalaxy.com.tw

PUBLISHER: Candy C. H. Lin

International Journal of Network Security

Vol. 20, No. 2 (Mar. 1, 2018)

A Secure Group Signature Scheme

Cheng-Yi Tsai[1], Pi-Fang Ho[2], Min-Shiang Hwang[1,3]

(Corresponding author: Min-Shiang Hwang)

Department of Computer Science and Information Engineering, Asia University[1]
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung 41354, Taiwan
(Email: mshwang@asia.edu.tw)
Department of Information Management, Chaoyang University of Technology[2]
Department of Medical Research, China Medical University Hospital, China Medical University[3]
No.91, Hsueh-Shih Road, Taichung 40402, Taiwan

(Invited Mar. 12, 2017)

Abstract

Group signature scheme could be applied to the valid members to represent the group. The validity of the signature could be verified by the receiver. On the other hand, the member who signs the message could not be found. However, the group manager could reveal and identify the signer if it is necessary. Concerning with a high performance on security, a new group signature scheme based on a discrete logarithm problem to achieve the characteristics of group signatures is proposed. With this proposed scheme, the signature could be generated rapidly. Also, the verification procedure of the group signature could be spent in a short time. This group signature scheme can protect important messages. Compare with other schemes, the proposed scheme is more secure and efficient than others. The analysis of the security and the performance evaluation of the proposed scheme are provided. The proposed group signature scheme could be suitable for e-commerce applications.

Keywords: Authenticated Encryption; Digital Signature; Group Signature

1 Introduction

A digital signature is used to prove the signed message that no non-valid person could sign [7, 38]. Anyone has the ability to verify the signature is signed or not by the signer. The basic requirements of the digital signature are the non-repudiation and unforgeable. No one can deny that he/she sign the message and forge a valid signature [9, 11, 30]. Furthermore, a group signature is a variation of the digital signature [4, 15, 16, 21, 25, 34, 37] that allows the valid member of the group to sign a message to represent the group. Hence, a group signature scheme has the following characters [2]. First, the valid members of the group could use the signature to sign messages. Second, the group signature could be verified. However, the exact signer could not be found. Finally, the identity of the signer could be revealed if it is necessary.

Based on the discrete logarithm problem, an efficient group signature scheme has been proposed [19]. However, some insecure questions in that scheme were pointed out [32]. Then, two improved group signature schemes were proposed by Tseng and Jan, respectively [32]. However, the proposed schemes did not satisfy the requirement of unlinkability and unforgeability [23,35]. Although there were some schemes proposed [3, 5, 19, 20, 26, 28, 29, 35], there exit insecure concerns. Based on the Diffie-Hellman technique, the contributory group key exchange protocol was proposed [33]. However, the protocol is weak to a man-in-the-middle attack [24]. Without bilinear pairings, an anonymous ID-based group key agreement protocol was proposed [14]. A group key agreement protocol based on braid groups which require only multiplication operations was provided [8].

In this paper, based on the discrete logarithm problem, the group signature scheme is proposed with the secure and efficient concerns. With an authenticated encryption, the signer might generate the signature for a message. The signed message could be recovered by the only specified receiver to verify. The concept of the encryption scheme desires to achieve the authenticity, the confidentiality, the integration, and the property of non-repudiation. Therefore, if the message with a group signature belongs to an important message, it is not expected to let unrelated others learn. By the way, a group signature has to be generated at the first step. Then, it encrypts the group signature and the relative message. In order to achieve this goal, this work proposes a group signature scheme based on authenticated encryption. It is expected to generate the group signature and the ciphertext simultaneously. The validity of the group signature could be verified and the encrypted message could be recovered [6, 13].

Hence, the expected secure group signature scheme has to meet the characteristics of correction, unforgeability, anonymity, unlinkability, exculpability, traceability and Coalition-resistance [1, 10, 12, 27]. The signature generated by the group member must be accepted by verification process. The only valid members in the group have an ability to sign the messages on behalf of the group. To find the exact signer is difficult within the computing sense. However, it could be revealed by the group manager. Besides, it is hard to tell if the two different signatures have been computed by the same member. The only valid member could use the signature on behalf of the belonged group. The group manager could identify the valid member to use the signature. Moreover, the group member of a colluding subset could not generate a valid signature. With the mentioned above, the secure group signature scheme is developing in this paper.

The following section describes the proposed scheme. The performance and the security analysis of the proposed scheme are shown in Section 3 and Section 4, respectively. Finally, the conclusion is given in Section 5.

2 Discrete Logarithm Problem

Based Scheme Based on discrete logarithm problem [17, 18, 22, 31], the group signature scheme is proposed in this paper. The proposed scheme includes three portions, initial phase, generation and verification, and identification.

2.1 Initiation Phase

Let p and q be two large primes such that $q|p-1$, and let g be a generator with order q in $GF(p)$. Each group member U_i selects a secret key x_i and computes the public key $y_i = g^{x_i} \bmod p$. The group manager T has the secret key x_T and the public key $y_T = g^{x_T} \bmod p$. For each group member U_i, the group manager randomly chooses an integer k_i in Z_q* and computes $r_i = y_i k_i - x_T \bmod q$ and $s_i = y_i k_i \bmod p$. Then, the group manager sends (r_i, s_i) to the group member U_i discreetly. After receiving (r_i, s_i), U_i may verify the validity by checking the equation $s_i y_i = (g^{r_i} y_T)^{x_i} \bmod p$.

2.2 Generation and Verification

A group member U_i signs the message M with the following steps,

1) choose two random numbers R_1, R_2 in Z_q*.

2) Compute A, B, C, and D as follows:

$$\begin{aligned} A &= x_i \cdot R_1 \cdot R_2 \bmod q. && (1)\\ B &= h^{-1}(M||A||D) g^{-R_1 \cdot A \cdot h(M||A||D)} \bmod p. \\ C &= g^{R_1 - r_i \cdot h(B)} \bmod p. \\ D &= s_i^{R_1 \cdot R_2 \cdot y_i} \bmod p. && (2)\end{aligned}$$

where $h()$ and $||$ denote a collision-resistant hash function and a concatenation, respectively.

3) The group signature becomes $\{A, B, C, D, M\}$.

The verification to the group signature is hold with the following the equation,

$$[Bh(M||A||D)]^{-e} \overset{?}{=} [C^A ({y_T}^{-A} D)^{h(B)}]^{Ch(M||A||D)} \bmod p. \quad (3)$$

2.3 Identification

The signature has to be revealed to identify the signer if it is needed. The group manager accesses the (y_i, k_i) of each member U_i, it require all (y_i, k_i)s to satisfying the following equation:

$$D == g^{A k_i y_i} \bmod p, \quad \text{for} \quad i = 1, 2, \cdots, n, \qquad (4)$$

where n is the number of group members. By the way, the group manager could determine who the signer is.

3 Performance Evaluation

The complexity of computing time is usually employed for the performance evaluation of the proposed scheme. In this work, some notations are used for convenience.

1) T_h denotes the time used for executing the one-way hash function $h()$.

2) T_{exp} is the time to execute a modular exponentiation operation.

3) T_{Nmul} is the time for multiplication with modulo N.

In the proposed group signature scheme based on a discrete logarithm problem, the signer requires $3T_{exp} + 8T_{Nmul} + 2T_h$ to generate a group signature. The verifier requires $5T_{exp} + 4T_{Nmul} + 2T_h$ to verify the group signature. Compared with our scheme and other schemes, the proposed scheme is better than that of the others schemes in performance.

4 Security Analysis

Based on the difficulty of the discrete logarithm problem, the security analysis to the proposed scheme is provided. The proposed scheme should meet all the security properties requests.

Correctness.

The receiver could verify the group signature $\{A, B, C, D, M\}$ by Equation (3).

Unforgeability and Exculpability.

A valid group signature could be generated by the valid membership (r_i, s_i) and the corresponding secret key x_i. In the case, the eavesdropper intercepts a valid member-ship (r_i, s_i) and intends to forge a group signature. According to the proposed scheme, he has to compute the parameters A, B, C and D

from Equation (1) to Equation (2). Without the secret key x_i, the eavesdropper could not forge a group signature. Either, Equation (3) could not be hold.

Anonymity.

Since the group signature scheme is designed for the group manager to identify the exact signer, all confidential information is protected by random parameters. Within a valid group signature $\{A, B, C, D, M\}$, A and D relates the identity information. Hence, the anonymity of A and D should be examined. With a valid group signature, Equation (1): $A = x_i \cdot R_1 \cdot R_2 \bmod q$,

$$g^A = g^{x_i \cdot R_1 \cdot R_2} = y_i^{R_1 \cdot R_2} \bmod p, \tag{5}$$

where R_1 and R_2 are integers. If R_1 and R_2 are known, y_i could be found, i.e. the exact signer could be identified. However, since the number R_1 and R_2 are unknown, no one could find the exact signer, i.e. the proposed scheme has anonymity.

Unlinkability.

Similarly to anonymity, to identify whether the signatures $\{A, B, C, D, M\}$ and the signature $\{A', B', C', D', M'\}$ are generated by the same group member is difficult. With Equations (3) and (4), the modified equations is given as the following,

$$g^A / g^{A'} = g^{x_i \cdot R_1 \cdot R_2} / g^{x_i \cdot R_1' \cdot R_2'} \bmod p \tag{6}$$

and

$$\begin{aligned} D/D' &= s_i^{y_i \cdot R_1 \cdot R_2} / s_i^{y_i \cdot R_1' \cdot R_2'} \\ &= \left(g^{x_i \cdot R_1 \cdot R_2} / g^{x_i \cdot R_1' \cdot R_2'}\right)^{k_i \cdot y_i} \bmod p. \end{aligned} \tag{7}$$

If one desired to check whether the two signatures are generated by the same signer, the equation

$$\left(g^A / g^{A'}\right)^{k_i \cdot y_i} = D/D' \bmod p, \tag{8}$$

should be hold. However, k_i and y_i are unknown. No one could determine whether the two group signatures are generated by the same signature.

Traceability.

The group manager could access the (y_i, k_i) for each member U_i. Hence, the group manager can request the (y_i, k_i) of U_i to meet the requirement in Equation (4). For the traceability, the group manager can determine the exact signer.

Coalition-resistance.

The group manager generates the (r_i, s_i) with the secret key x_T for each group member. Then, the group manager sends (r_i, s_i) to the group member j. If the colluding subset of the group members desires to generate a valid group signature, they have to keep the secret key x_T. However, the colluding subset of group members does not keep the secret key. The valid (r_i, s_i) could not be forged. Hence, a valid group signature could not be generated. The group manager could not link to any member in the colluding group.

Based on the above security analysis of the proposed scheme, it is shown the proposed scheme could approach all security property requirements.

5 Conclusions

Group signature scheme functions to protect the important messages. In this paper, a new group signature scheme based on discrete logarithm problem has been proposed. The performance and security analysis are given to show the proposed scheme has a superior capacity. With the proposed scheme, the signers could generate a group signature swiftly, and the verification could be quickly complemented. For the applications with the time efficiency concern, the proposed scheme could be employed in the e-commerce.

Acknowledgment

This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: MOST 104-2221-E-468-004 and MOST 105-2410-H-468-009.

References

[1] B. E. Ayebie, H. Assidi, El M. Souidi, "A new dynamic code-based group signature scheme," *Lecture Notes in Computer Science*, vol. 10194, pp. 346-364, Springer-Verlag, 2017.

[2] N. Begum, T. Nakanishi, S. Sadiah, Md. E. Islam, "Implementation of a revocable group signature scheme with compact revocation list using accumulator," in *4th International Sym-posium on Computing and Networking (CANDAR'16)*, pp. 610-615, 2017.

[3] L. Boongasame, P. Temdee, F. Daneshgar, "A group signature based buyer coalition scheme with trustable third party," *International Journal of Production Research*, vol. 55, no. 17, pp. 5050-5061, 2017.

[4] D. Chaum and E. Heyst, "Group signatures," in *Advances in Cryptology (Eurocrypt'91)*, pp. 257-265, 1991.

[5] E. M. Cho, T. Koshiba, "Secure deduplication in a multiple group signature setting," *Proceedings of International Conference on Advanced Information Networking and Applications (AINA'17)*, pp. 811-818, 2017.

[6] H. Ge, "An effective method to implement group signature with revocation," *International Journal of Network Security*, vol. 5, no. 2, pp. 134-139, 2007.

[7] M. Hassouna, E. Bashier, and B. Barry, "A strongly secure certificateless digital signature scheme in the

random oracle model," *International Journal of Network Security*, vol. 18, no. 5, pp. 938-945, 2016.

[8] P. Hiranvanichakorn, "Provably authenticated group key agreement based on braid groups - The dynamic case," *International Journal of Network Security*, vol. 19, no. 4, pp. 517-527, 2017.

[9] M. S. Hwang and C. C. Lee, "Research issues and challenges for multiple digital signatures," *International Journal of Network Security*, vol. 1, no. 1, pp. 1-7, 2005.

[10] M. H. Ibrahim, "Resisting traitors in linkable democratic group signatures," *International Journal of Network Security*, vol. 9, no. 1, pp. 51-60, 2009.

[11] A. U. Khan and B. K. Ratha, "A secure strong designated verifier signature scheme," *International Journal of Network Security*, vol. 19, no. 4, pp. 599-604, 2017.

[12] S. Khomejani and A. Movaghar, "Privacy consideration for trustworthy vehicular ad hoc networks," in *2010 International Conference On Electronics and Information Engineering*, pp. 437-442, 2010.

[13] K. Kim, I. Yie, S. Lim, and D. Nyang, "Batch verification and finding invalid signatures in a group signature scheme," *International Journal of Network Security*, vol. 13, no. 2, pp. 61-70, 2011.

[14] A. Kumar and S. Tripathi, "Anonymous ID-based group key agreement protocol without pairing," *International Journal of Network Security*, vol. 18, no. 2, pp. 263-273, 2016.

[15] C. C. Lee, T. Y. Chang, M. S. Hwang, "A new group signature scheme based on the discrete logarithm," *Journal of Information Assurance and Security*, vol. 5, no. 1, pp. 54-57, 2010.

[16] C. C. Lee, P. F. Ho, M. S. Hwang, "A secure E-auction scheme based on group signatures," *Information Systems Frontiers*, vol. 11, no. 3, pp. 335-343, July 2009

[17] C. C. Lee, M. S. Hwang, L. H. Li, "A new key authentication scheme based on discrete logarithms", *Applied Mathematics and Computation*, vol. 139, no. 2, pp. 343-349, July 2003.

[18] C. C. Lee, M. S. Hwang, and W. P. Yang, "A new blind signature based on the discrete logarithm problem for untraceability", *Applied Mathematics and Computation*, vol. 164, no. 3, pp. 837-841, May 2005.

[19] W. B. Lee and C. C. Chang, "Efficient group signature scheme based on the discrete logarithm," *IEE Proceedings - Computer Digital Technology*, vol. 145, no. 1, pp. 15-18, 1998.

[20] L. H. Li, C. Y. Liu, and M. S. Hwang, "Cryptanalysis of an efficient secure group signature scheme," *ACM Operating Systems Review*, vol. 38, no. 4, pp. 67-69, 2004.

[21] L. H. Li, S. F. Tzeng, M. S. Hwang, "Generalization of proxy signature based on discrete logarithms," *Computers & Security*, vol. 22, no. 3, pp. 245-255, 2003.

[22] L. H. Li, S. F. Tzeng, M. S. Hwang, "Improvement of signature scheme based on factoring and discrete logarithms", *Applied Mathematics and Computation*, vol. 161, no. 1, pp. 49-54, Feb. 2005.

[23] Z. Li, L. C. K. Hui, K. P. Chow, C. F. Chong, W. W. Tsang, and H. W. Chan, "Security of Tseng-Jan's group signature schemes," *Information Processing Letters*, vol. 75, no. 5, pp. 187-189, 2000.

[24] C. H. Ling, S. M. Chen, and M. S. Hwang, "Cryptanalysis of Tseng-Wu group key exchange protocol," *International Journal of Network Security*, vol. 18, no. 3, pp. 590-593, 2016.

[25] E. J. L. Lu, M. S. Hwang, and C. J. Huang, "A new proxy signature scheme with revocation," *Applied Mathematics and Computation*, vol. 161, no. 3, PP. 799-806, Feb. 2005.

[26] K. M. Nomura,S. Masami, M. M. Yoshiaki, "A multigroup signature scheme for local broadcasting," *14th IEEE Annual Consumer Communications and Networking Conference (CCNC'17)*, pp. 449-454, 2017.

[27] F. M. Salem, M. H. Ibrahim, and I. I. Ibrahim, "Non-interactive authentication scheme providing privacy among drivers in vehicle-to-vehicle networks," in *Sixth International Con-ference on Networking and Services*, pp. 156-161, 2010.

[28] Z. Shao, "Repairing efficient threshold group signature scheme," *International Journal of Network Security*, vol. 7, no. 2, pp. 218-222, 2008.

[29] R. H. Shi, "An efficient secure group signature scheme," in *Proceedings of IEEE (TENCON'02)*, pp. 109-112, 2002.

[30] N. Tiwari and S. Padhye, "Provable secure multiproxy signature scheme without bilinear maps," *International Journal of Network Security*, vol. 17, no. 6, pp. 736-742, 2015.

[31] C. Y. Tsai, C. Y. Liu, S. C. Tsaur, and M. S. Hwang, "A publicly verifiable authenticated encryption scheme based on factoring and discrete logarithms", *International Journal of Network Security*, vol. 19, no. 3, pp. 443-448, May 2017.

[32] Y. M. Tseng and J. K. Jan, "Improved group signature scheme based on discrete logarithm problem," *IEE Electronics Letters*, vol. 35, no. 1, pp. 37-38, 1999.

[33] Y. M. Tseng and T. Y Wu, "Analysis and improvement on a contributory group key exchange protocol based on the Diffie-Hellman technique," *Informatica*, vol. 21, no. 2, pp. 247-258, 2010.

[34] S. F. Tzeng, M. S. Hwang, "Digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem," *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 61-71, Mar. 2004.

[35] G. Wang, "Security analysis of several group signature schemes," *Lecture Notes in Computer Science*, vol. 2904, pp. 252-265, Springer, 2003.

[36] Y. Wang J. Zhang, X. Chen, "Security analysis of the improved group signature," in *Information Theory Workshop*, pp. 171-174, 2003.

[37] C. C. Yang, T. Y. Chang, M. S. Hwang, "A new group signature scheme based on RSA assumption," *Information Technology and Control*, vol. 42, no. 1, pp. 61-66, 2013.

[38] X. Zhang, R. Lu, H. Zhang, and C. Xu, "A new digital signature scheme from layered cellular automata," *International Journal of Network Security*, vol. 18, no. 3, pp. 544-552, 2016.

Biography

Cheng-Yi Tsai received his B.S. degree from Department of Business Administration, Chaoyang University of Technology (CYUT), Taiwan in 2001 and M.S. degree from Computer Science & Information Engineering, Asia University, Taiwan in 2005 . He is currently pursuing the Ph.D. degree from Department of Computer Science and Information Engineering, Asia University, Taiwan. His research interests include blockchain, information security, and cloud computing.

Pi-Fang Ho received the B.S. in Applied Mathematics, Providence University, Taiwan, in 2003; the M.S. in Information Management, Chaoyang University of Technology, Taiwan, in 2005. Her current research interests include communication and information security.

Min-Shiang Hwang received Ph.D. degree in computer and information science from the National Chiao Tung University, Taiwan in 1995. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories, Ministry of Transportation and Communications. He was also the Chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2002. He was a professor and Chairman of the Department of Management Information Systems, NCHU, during 2003-2009. He was also a visiting professor with University of California (UC), Riverside and UC. Davis (USA) during 2009-2010. He was a distinguished professor of the Department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Excellent Research Award of National Science Council (Taiwan). Dr. Hwang was a dean of College of Computer Science, Asia University (AU), Taichung, Taiwan. He is currently a chair professor with the Department of Computer Science and Information Engineering, AU. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include information security, electronic commerce, database and data security, cryptography, steganography, and mobile computing. Dr. Hwang has published over 200+ articles on the above research fields in international journals.

Robust Speech Perception Hashing Authentication Algorithm Based on Spectral Subtraction and Multi-feature Tensor

Yi-Bo Huang[1], Qiu-Yu Zhang[2], Wen-Jin Hu[2]

(Corresponding author: Yi-Bo Huang)

College of Physics and Electronic Engineering, Northwest Normal University[1]
No.967, An-Ning East Road, Lanzhou 730070, China
(Email: huang_yibo@foxmail.com)
School of Computer and Communication, Lanzhou University of Technology[2]
No.287, Lan-Gong-Ping Road, Lanzhou 730050, China

(Received Apr. 20, 2017; Revised and Accepted Aug. 26, 2017)

Abstract

In order to make the speech perception hashing authentication algorithm has strong robustness and discrimination to content preserving operations and speech communication under the common background noise, a new robust speech perceptual hashing authentication algorithm based on spectral subtraction and multi-feature tensor was proposed. The proposed algorithm uses spectral subtraction method to denoise the speech which processed by applying pre-processing. Then, the algorithm acquires each speech component wavelet packet decomposition, MFCC and LPCC feature of each speech component are extracted to constitute the speech feature tensor. The feature tensor is decomposed tensor decomposition to reduce the complexity. Finally, speech authentication is done by generating the hashing values which use mid-value. Experimental results show that the proposed algorithm can denoise the speech effectively, and have good robustness and discrimination to content preserving operations, as well as able to resist the attack of the background noise, which is commonly heard during the communication.

Keywords: Background Noise; Multi-feature Tensor; Robust; Spectral Subtraction; Speech Perceptual Hashing

1 Introduction

Speech signal is easily to be disturbed in the transmission channel, in the speech instant messaging; the speech is usually affected by coding and decoding [17], channel noise, delay, packet loss, and the impact of the retrieval speed. In order to achieve efficient speech authentication, how to solve the problem of the interaction between robustness, distinguish and authentication efficiency, so it is very important to study the speech noise reduction technology and speech perceptual hashing authentication [2].

Therefore it is necessary to consider whether the speech feature can be extracted completely and accurately, and it is required that the calculation of the perceptual speech hashing robust should be the strongest, the coupling should be minimum, calculating should be easy. The extraction of the speech perception feature value is the key of speech perceptual authentication. In order to reduce the influence of noise on speech feature extraction, speech denoising technique is used in preprocessing. At present, the speech noise reduction methods mainly include: spectral subtraction, Wiener filtering method, Kaman filtering method, adaptive filtering method and so on. The current extraction of the speech perception feature is based on the human ear psycho acoustic model, the speech perceptual hashing feature value extraction and proceeding methods mainly include: the spectrum coefficient [11], linear predictive coding (LPC) [9], Mel-scale Frequency Cepstral Coefficients (MFCC) [4, 6] line spectrum frequency (LSF) [10], Energy to Entropy Ratio [19], frequency cepstral coefficients [12], Hilbert transform [21] and bark-bands energy [14]. The literature [4] proposed a Speech perception hashing algorithm that based on the Mel-scale Frequency Cepstral Coefficients and Nonnegative matrix factorization (NMF), the paper proposed a singular value decomposition then obtains the speech information, and then undergo the NMF. It reduces the mistakes and gives out a satisfactory outcome of the hashing function. The experiment shows that the Robustness is improved but the Distinction is poor, due to the principal component analysis method is used in the algorithm, the time complexity of the algorithm is large and it cannot meet the requirements of realtime speech authentication. Chen [3] proposed a speech perceptual hashing algorithm based on LPC combined with non-negative matrix factorization.

The algorithm has good ability of collision resistance, but it is not effective to distinguish the different speeches and content preserving operations. Zhang [22] proposed an efficient speech perception hashing algorithm based on a linear predictive residual coefficient of LP analysis combined with G.729 coding. The algorithm has good robustness, discrimination and high efficiency, but robustness is poor when the signal noise ratio is low. Li [8] proposed a speech perception hashing algorithm based on MFCC correlation coefficients combined with pseudo random sequences, the algorithm has good robustness, discrimination and security, but collision resistance is poor and performance at the low signal noise ratio is not good.

In order to solve the problem of robustness and discrimination in speech perception hashing authentication, we present a robust perceptual hashing based on spectral subtraction and multi-feature tensor after analyze the data that used spectral subtraction and without applying spectral subtraction. The proposed algorithm can solve the problem of the mutual influence between the robustness of content preserving operations, discrimination and authentication efficiency. Firstly, preprocessing of the speech signal used spectral subtraction to denoise the speech signal noise. Secondly, introduces the method of MFCC coefficients and LPC cepstrum coefficients in the process of perception speech hashing, feature modeling based on multi-feature, Construction of feature tensor by multi-feature, finally, the authentication function is realized by using tensor decomposition and hashing structure.

The rest of this paper is organized as follows. Section 2 describes the basic theory of spectral subtraction for noise reduction and the basic algorithm of multi-feature. A detailed speech perceptual hashing authentication scheme is described in Section 3. Section 4 gives the experimental results as compared with other related method. Finally, we conclude our paper in Section 5.

2 Problem Statement and Preliminaries

2.1 Spectral Subtraction for Noise Reduction

The spectral subtraction speech enhancement is utilized broadly because it is simple and easy for the realtime processing [23]. The main idea of spectral subtraction is the independence of noise and speech signal, it will be Noisy speech power spectrum minus the noise power spectrum, and then get the pure speech spectrum.

$$y(t) = x(t) + n(t).$$

Let $x(t)$ be a speech signal, $n(t)$ is a noise signal, and $y(t)$ is a noisy speech signal.

$$Y(\omega) = S(\omega) + N(\omega). \tag{1}$$

Equation (1) is a frequency expression.

$$\begin{aligned} |Y(\omega)|^2 &= |S(\omega)|^2 + |N(\omega)|^2 + 2Re[S(\omega)N*(\omega)] \\ E(|Y(\omega)|^2) &= E(|S(\omega)|^2) + E(|N(\omega)|^2) \\ &\quad +2E(Re[S(\omega)N*(\omega)]). \end{aligned} \tag{2}$$

In Equation (2), $S(\omega)$ and $N(\omega)$ are completely independent. $N(\omega)$ submit to zero mean value normal distribution. Equation (3) can be written as:

$$|Y(\omega)|^2 = |S(\omega)|^2 + |N(\omega)|^2 \tag{3}$$

$N(\omega)|^2$ can be estimated by Silent section. The estimated value of the original speech is defined as in Equation (4):

$$|S(\omega)| = [|Y(\omega)|^2 - |N(\omega)|^2]^{\frac{1}{2}} \tag{4}$$

2.2 LPCC Feature Coefficient Extraction

LPCC is a commonly used speech feature. This feature can be used to build the speech model, and the speech model is considered as the all pole model, which can be realized simply and easily in algorithm. But for the voiceless and nasal recognition effect is poor. We can directly derive the cepstrum from the linear prediction coefficient. The recurrence relation between LPC coefficient and LPCC coefficient is below:

$$\begin{aligned} c_0 &= a_1 \\ c_n &= a_n + \sum_{k=1} n - 1\frac{k}{n}c_k a_{n-k}, 1 \leq n \leq N \\ c_n &= \sum_{k=1} n - 1\frac{k}{n}c_k a_{n-k}, n > N. \end{aligned}$$

Here, c_0 is the DC component, a_n is the LPC coefficient, and c_n is the LPCC coefficient.

2.3 MFCC Feature Coefficient Extraction

When speech feature are extracted, the MFCC [13] is mostly used as the feature vector [5]. Mel scale describes the nonlinear feature of human ear's frequency perception. Its relation with the practical frequency of speech. The equation below:

$$Mel(f) = 2595\log(1 + \frac{f}{700}), 0 \leq f \leq F_n.$$

In the equation above, f is the practical speech frequency; F_n is the Nyquist frequency of speech signal.

2.4 Wavelet Packet Transform

Discrete wavelet transform (DWT) has the ability to accurately characterize local details of speech signals [1, 20] Wavelet packet transform (WPT) as the further expansion of wavelet analysis theory. Wavelet packet decomposition can reflect the feature and nature of the signal. It is very suitable for the analysis and processing of

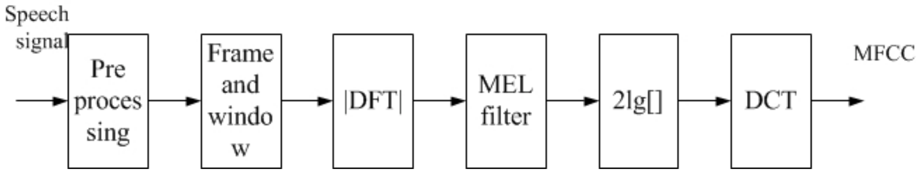


Figure 1: MFCC feature extraction process

speech signal types of non-stationary signal [15, 16, 18]. K level wavelet packet decomposition principle is shown in Figure 2. The subspace U_i^m is $U_m(t)$ and $U_{2m}(t)$'s closure spaces, speech signal through the recursive equation wavelet packet decomposition:

$$\begin{aligned} u_{2m}(t) &= \sqrt{2}\sum_{n\in Z} h(n)\mu_m(2t-n) \\ u_{2m+1}(t) &= \sqrt{2}\sum_{n\in Z} g(n)\mu_m(2t-n). \end{aligned}$$

Here, $h(n)$ is a high pass filter group, and $g(n)$ is a low pass filter group, $g(n) = (-1)^n h(1-n)$, and that the two coefficients with orthogonal relation.

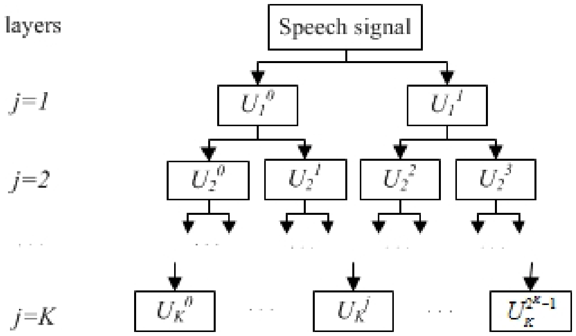


Figure 2: The decomposition graph of K-level wavelet packet

3 The Proposed Scheme

3.1 Establishment of Speech Tensor Model

Tensor can be considered as a product of vector space, and it is a higher order generalization of vector and matrix. The order of the tensor can be expressed as $X \in R^{N_1 \times N_2 \times L \times N_M}$. Tensor decomposition method is widely used in image processing, pattern recognition, data compression and so on [7]. It can better show the relationship among speech frame structure, decomposition scale and feature coefficient. Figure 3 shows the schematic diagram for the construction of speech tensor, which can directly describe the structure of speech tensor.

Describe the speech feature from three perspectives, which are respectively the speech frame, wavelet packet decomposition scale as well as MFCC and LPCC feature coefficients. The speech frame mainly describes the precedence relationship of speech and describes the relationship of speech feature from the time scale. The wavelet package decomposition scale conducts wavelet package decomposition for each frame of speech signal so as to different scales of approximate components and detailed components of each frame of speech signal. MFCC and LPCC feature coefficients conduct the feature extraction for the components decomposed by each wavelet package to obtain the component feature. Tensor construction can be carried out for a section of speech from the above three perspectives. The tensor constructed is the third order speech tensor of [speech frame × wavelet package decomposition scale × MFCC and LPCC feature coefficients].

Figure 4 shows the construction method for speech feature tensor adopted in this paper. The construction diagram consists of the speech signal preprocessing, speech feature extraction and speech feature tensor construction.

The Tucker decomposition model is the product of N-order tensor $X \in R_{I_1 \times I_2 \times L \times I_n}$ through Tucker decomposition to obtain the product of a lower dimensional core tensor G and N projection matrix $U^{(n)}$. Tucker decomposition model is below:

$$X \approx G \times_2 U^{(1)} \times_2 U^{(2)} \times L \times_N U^{(N)}.$$

$G \in R_{J_1 \times J_2 \times L \times J_N}$ is core tensor. The main information of the original tensor is retained in the core tensor. $U^{(n)} \in R_{I_n \times J_n}$ is projection matrix, $J_n \leq I_n$ and $U^{(n)}$ are orthogonal. Tucker decomposition can be used optimal decomposition to solve the optimization problem.

$$\begin{aligned} \min |X - G \times_1 U^{(1)} \times_2 U^{(2)} \times L \times_N U^{(N)}|^2, \\ G \geq 0, U^{(n)} \geq 0. \end{aligned}$$

If $J_n = rank_{(n)}X$, Tucker decomposition is meaningless. If $J_n < rank_{(n)}X$, Tucker decomposition is meaningful.

3.2 Quantization

Reconstruct the core tensor G to form the two-dimensional feature matrix. Calculate the sum of each column of matrixes:

$$R_h(j) = \sum_{i=1}^{r} H_{ij}^{(n)}, 1 \leq j \leq k.$$

In the equation above, $H_{ij}^{(n)}$ signifies the feature coefficient in row j and line I, k is the number of rows of feature

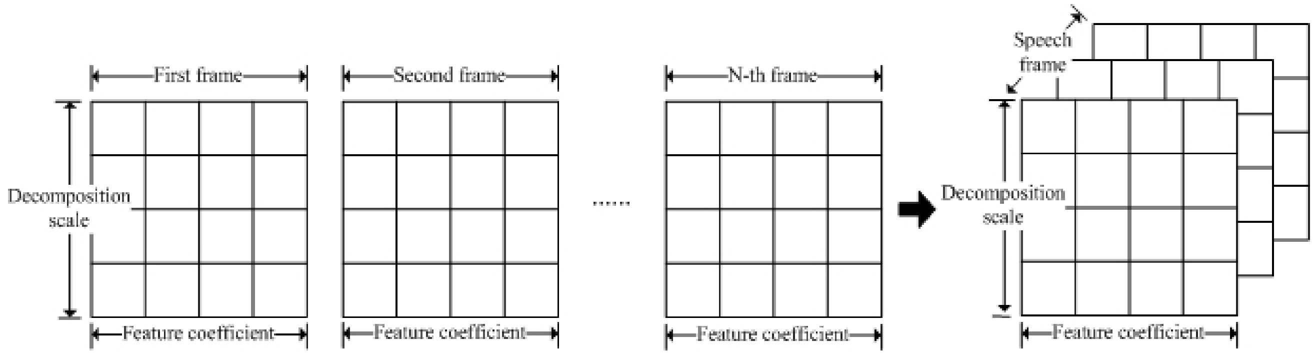


Figure 3: The structure graph of speech tensor

matrix. Quantize the coefficient formed and row matrix to form the hashing value $h(j)$ of speech segment;

$$h(j) = \left\{ \begin{array}{ll} 1 & R_h(j) > \hat{R}_h, 1 \leq j \leq k \\ 0 & \text{other} \end{array} \right\}$$

In the equation above, $\hat{R_h}$ is the mid-value.

3.3 Speech Perception Hashing Authentication Scheme

Figure 4 describes the construction of speech tensor. After the tensor decomposition, since the core tensor G is less than the original tensor X, the core tensor G can be considered as the compression form of original tensor X. In this algorithm, the core tensor G is used to describe the speech feature. The flow chart of speech perception hashing authentication based on Spectral Subtraction and Multi-feature tensor is shown in Figure 5.

The detailed steps of the algorithm are shown below:

Step 1: Preprocessing: conduct pre-emphasis on the speech in the speech library to be tested, enhance the useful frequency spectrum of high frequency, reduce the edge effect and eliminate noise.

Step 2: Spectral subtraction for noise reduction: the speech signal is processed by spectral subtraction, In the spectral subtraction experiment, the length of frame is 30ms, frame shift is 25ms, $NIS = 8$, $a = 3$, $b = 0.5$.

Step 3: Framing and windowing: in order to eliminate the inter frame loss during framing, conduct framing and add the Hamming window for speech $x(t)$; during framing, the frame length is L; when the frame moves at $L/2$, $s(n)$ can be obtained; later, add Hamming window for $s(n)$ to obtain $s_w(n)$, n is the frame number.

Step 4: Wavelet packet transform: carry out wavelet package decomposition for the speech frame. In this paper, the 3-order wavelet packet decomposition is carried out, 8 speech segments are obtained, and then calculate the MFCC coefficient and LPCC coefficient of each segment.

Step 5: Construction and decomposition of speech tensor: conduct the tensor construction of feature coefficient to obtain the speech feature tensor X, carry out Tucker decomposition for the feature tensor X to obtain the low-dimensional core tensor G and the project matrix $U^{(n)}$.

Step 6: Quantization: construct the core tensor G and thus obtain the sequence $R_h(j)$; quantize $R_h(j)$ to obtain the perception hashing sequence $h(j)$;

Step 7: Calculation and matching of perception hashing distance: suppose that there are two speech segments α and β, define the hashing mathematic distance is $D_h(:,:)$, which is shown below:

$$D_h(H_\alpha, H_\beta) = \sum |h_\alpha(j) - h_\beta(j)|, j = 1, 2, L, n.$$

Match according to the hypothesis testing of hashing mathematic distance $D_h(:,:)$ and hashing sequence $h(:)$ as follows:

K1: If the perception contents of the two speech segments α and β are the same:

$$D_h(H_\alpha, H_\beta) \leq \tau.$$

K2: If the perception contents of the two speech segments α and β are different:

$$D_h(H_\alpha, H_\beta) > \tau.$$

In the equations above, τ is matching threshold. The matching threshold can be used to determine whether the perception contents of speech signals are the same so as to realize the perception hashing authentication of speech signals.

4 Experimental Results and Analysis

The operating software environment is MATALB 2010b. The operating experimental hardware platform is Intel(R)

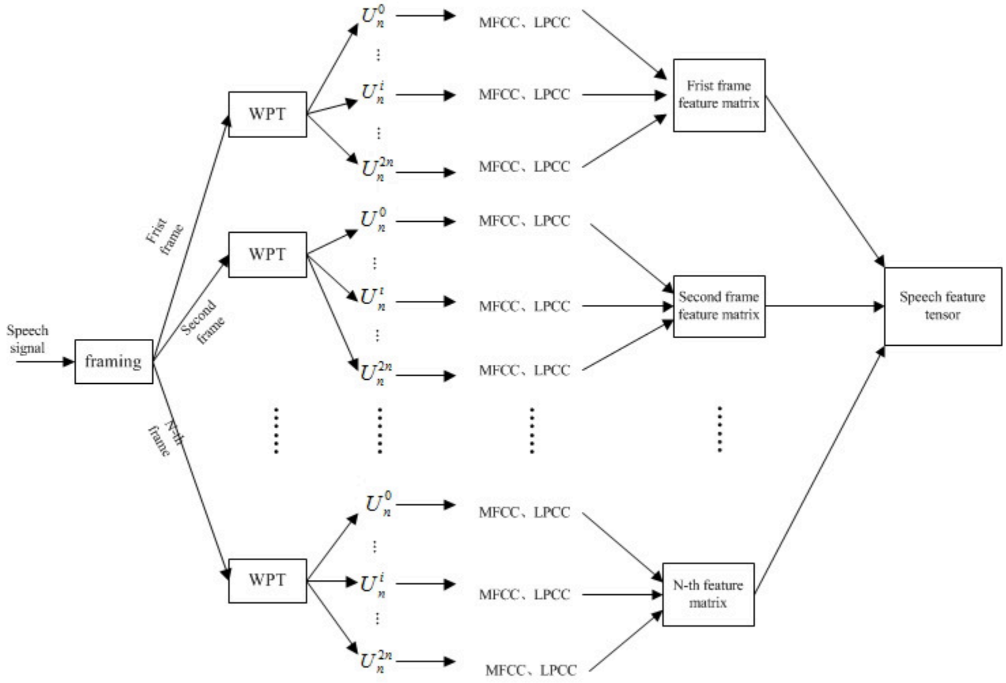


Figure 4: The structure graph of speech feature tensor

Core(TM) i5-4590 CPU 3.30GHz, with computer memory of 4GB. The speech data used in the experiment is the speech in the Texas instruments and Massachusetts institute of technology (TIMIT) speech library which is composed of different contents recorded English by men and women, and Noisex-92 noise library as noise library. The speech clip length is 4s. The speech library in this paper is a total of 1,280 speech clips. The content preserving operations are performed for the 600 speech clips, as shown in Table 1.

4.1 Discrimination Test and Analysis

Discrimination is mainly used to evaluate the reliability of the algorithm for distinguishing different speech contents read by different or same persons. Since the bit error rates (BER) of different speech segments are random variables, this experiment analyzes the discrimination of algorithm with the probability distribution curve. The BER of the perceptual hashing values of different speech contents basically obeys the normal distribution. By pairwise comparison of perceptual hash values for 600 speech clips, there are 179700 BER values are obtained. Compare every two of speech in the speech library and the diagram of BER normal distribution obtained is shown in Figure 6.

When the error rate is used as the distance measure, it should approximately abbey the normal distribution. It can be seen from Figure 6 that the probability curve of standard normal distribution overlaps the probability distribution of BER value of this algorithm, so the hashing distance obtained through this algorithm approximately obeys the normal distribution; namely, speech with different perceptions will generate different hashing values.

In the ideal condition, every speech segments with different contents will have its different perception hashing valve and every pair of hashing value matching should have a high error rate. Actually, there are always a few of BER data which are low and probably lower than the threshold value, then it will be wrongly judged as same content. According to Table 1, it can know that the false acceptation rate (FAR) increases with the enlargement of BER threshold value. Compared with the other two algorithms, the algorithm proposed in this paper has a strong collision resistance. When the threshold value $\tau = 0.25$, the collision probability is that 6 segments among 10^{10} speech segments may collide. When $\tau = 0.27$, the collision probability is that 1 segments among 10^8 speech segments may collide. When $\tau = 0.30$, the collision probability is that 6 segments among 10^7 speech segments may collide. It can be seen from Figure 6 that 2 segments among 10^5 speech segments will collide when the threshold value $\tau = 0.35$. As indicated in Table 1, compared with other two algorithms, the algorithm is very stable in the collision resistance. Therefore, this algorithm can correctly identify the authenticated speech segments.

The mean of matching between different speech is 0.4996 and the square is 0.0411, as shown in Figure 7, the μ and δ measured in the experiment are close to the theoretical results.

Table 1: Content preserving operation

Operating means	Operation method	Abbreviation
Volume Adjustment 1	Volume down 50%	V. ↓
Volume Adjustment 2	Volume up 50%	V. ↑
Resampling 1	Sampling frequency decreased to 8kHZ, and then increased to 16kHZ	R.8 → 16
Resampling 2	Sampling frequency decreased to 32kHZ, and then increased to 16kHZ	R.32 → 16
Narrowband noise 1	SNR=20db narrowband Gaussian noise, center frequency distribution in 0 4kHZ	G.N20
Narrowband noise 2	SNR=30db narrowband Gaussian noise, center frequency distribution in 0 4kHZ	G.N30
MP3 Compression 1	Re-encoded as MP3, and then decoding recovery, the rate is 32k	M.32
MP3 Compression 2	Re-encoded as MP3, and then decoding recovery, the rate is 48k	M.48
MP3 Compression 3	Re-encoded as MP3, and then decoding recovery, the rate is 128k	M.128
MP3 Compression 4	Re-encoded as MP3, and then decoding recovery, the rate is 192k	M.192

Table 2: The comparison results of FAR value

	[4]	Without applying spectral subtraction algorithm	The proposed algorithm
$tau = 0.20$	5.1875e-013	5.5635e-014	1.6299e-013
$tau = 0.25$	2.0985e-009	3.0135e-010	6.4908e-010
$tau = 0.27$	6.4075e-008	6.1722e-009	1.1930e-008
$tau = 0.30$	4.5267e-006	3.6602e-007	6.1098e-007
$tau = 0.32$	4.4124e-005	4.1395e-006	6.3326e-006
$tau = 0.35$	9.7400e-003	1.0136e-004	1.3820e-004

Table 3: The matching rate of speech authentication after the being kept operating content %

	[4]	Without applying spectral subtraction algorithm	The proposed algorithm
V. ↓	98.3	98.86	100
V. ↑	97.4	100	100
R.8 → 16	97.2	95.3	95.3
R.32 → 16	96.4	98.1	100
G.N20	78.1	86.7	91.6
G.N30	93.4	94.7	95.3
M.32	84.2	87.4	91.4
M.48	91.2	96.7	96.7
M.128	94.6	96.8	98.7
M.192	100	100	100

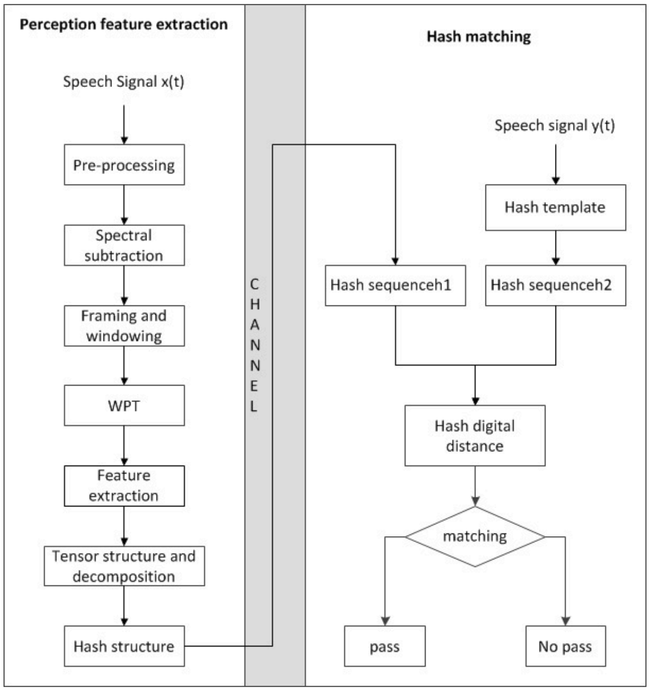


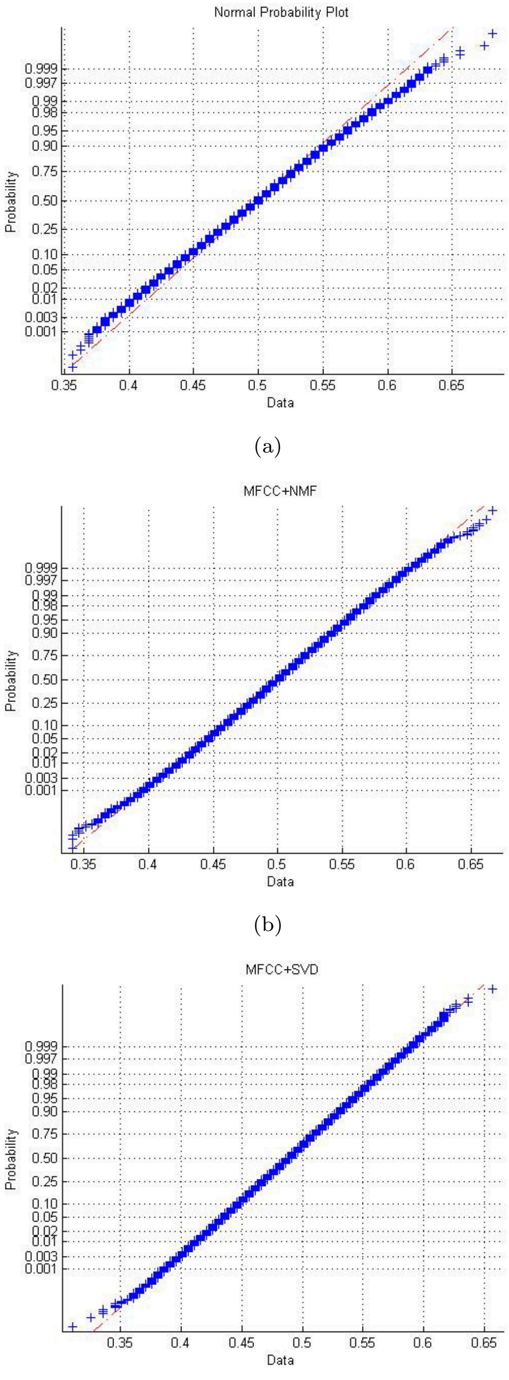
Figure 5: The flowchart of speech perception hashing authentication

4.2 Robustness Test and Analysis

The speech perception hashing robustness is mainly used to evaluate the reliability of the same speech after different preserving operations. The content preserving operations are performed for the 1280 speech clips, as shown in Table 1. The comparison results in various BER and the algorithm without applying spectral subtraction method are shown in Table 2.

As can be seen form Table 2, the proposed algorithm has good robustness for increasing and decreasing of the volume, filtering, resampling and re-encoding than that without applying spectral subtraction algorithm and [4], this is due to about content preserving operations have little effect on speech feature, so the algorithm has good robustness. However, the noise has great influence on the LPCC and MFCC coefficient, so the effect is not good on the speech added noise whether it is 20db or 30db. We can analyze the data from Table 3, when applying spectral subtraction method, we can see that the mean values of all content preserving operation are decrease, but the running efficiency is improved by nearly one times. It has a good improvement on the volume adjustment, resampling and Gaussian noise, this is because of the above operations have great influence on the speech amplitude and noise clip, so the effect is improved obviously by applying spectral subtraction method. According to the data in Table 2 and Table 3, as for the robustness of preserving content operation, the proposed algorithm is generally stronger than other two algorithms.

Based on the BER rate of content preserving operation, the false acceptance rate (FAR) and false reject rate (FRR) are obtained. Draw the FAR-FRR curve, which is shown in Figure 8.

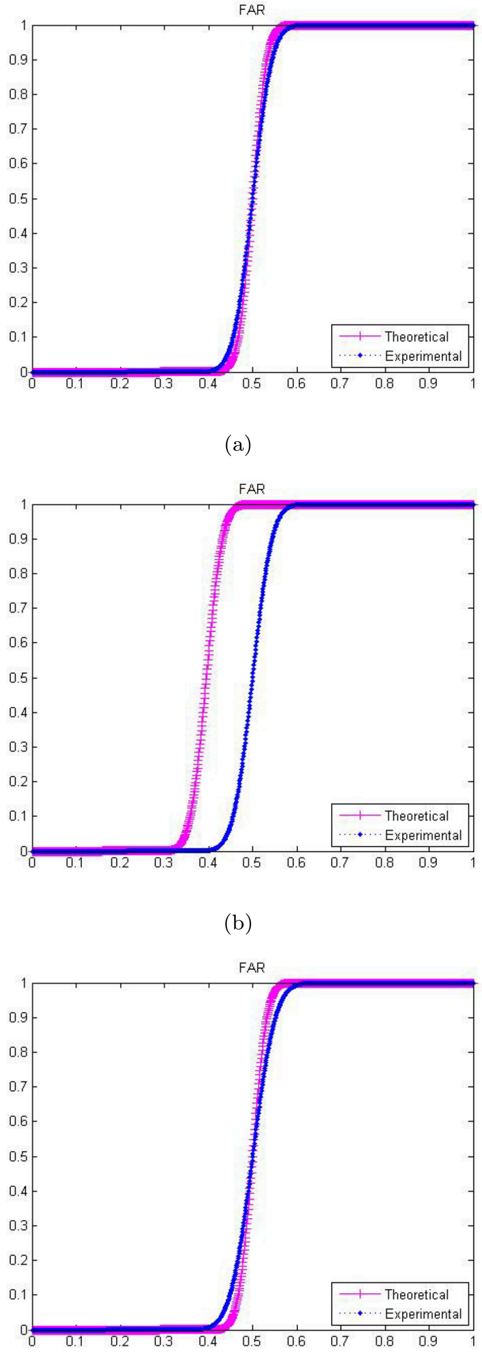


(a)

(b)

(c)

Figure 6: The BER normal distribution diagram. (a) The Proposed algorithm, (b) The algorithm of [4], (c) Without applying spectral subtraction algorithm.



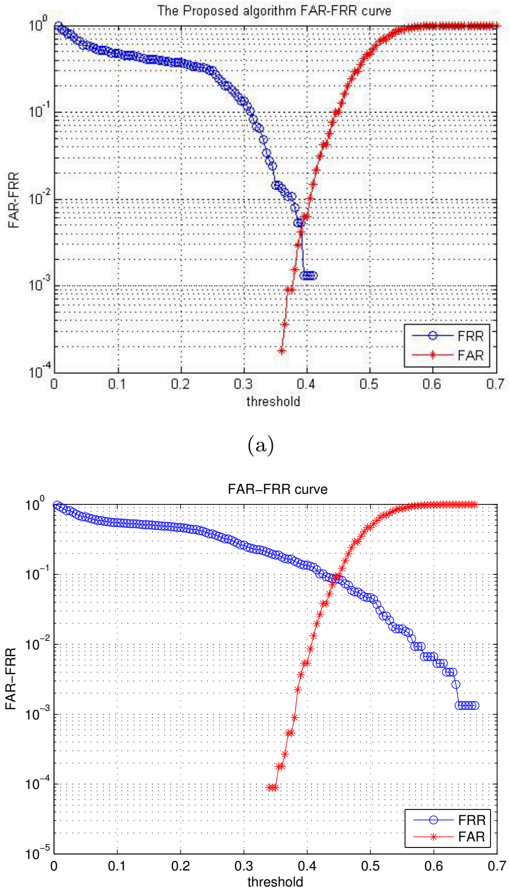
(a)

(b)

(c)

Figure 7: FAR curves of the algorithm. (a) The Proposed algorithm, (b) The algorithm of [4], (c) Without applying spectral subtraction algorithm.

This paper totally get 179,700 BER values by conducted pairwise comparison between perceptual hashing values form 600 different speech clips, and the false accept rate and false reject rate (FRR) is obtained via above attacks, and drawing the FAR-FRR curve, the results of comparison between without applying spectral subtraction method and the algorithm in [4] are shown in Figure 8. As indicated in the result in Figure 8, the FAR-FRR curve obtained by the proposed algorithm has a litter cross. The proposed algorithm's FRR curve does intersects with the FAR curve in 0.39, the FRR curve shows a significant convergence and a very wide judging domain; the judging threshold is between 0.35 and 0.40, showing a significant judging domain. Compared with the algorithms which without applying spectral subtraction algorithm, this algorithm has good robustness, and can correctly authenticate the same and different speech segments and, meanwhile, the algorithm authenticate the speech segments which go through the content holding operation and malicious attack. Therefore, compared with other two algorithms, this one has good discrimination and robustness.



(a)

(b)

Figure 8: The FAR-FRR curves of different perceptual hashing algorithm. (a) The Proposed algorithm, (b) Without applying spectral subtraction algorithm.

Table 4: BER mean and running time comparison

Operating	Mean	Variance	Max	Mean time	Mean	Variance	Max	Mean time
Algorithm	The proposed algorithm				Without applying spectral subtraction algorithm			
V. ↓	0.1019	0.3187	0.3187	72min40s	0.1253	0.3723	0.3723	65min42s
V. ↑	0.1121	0.3312	0.3312		0.0811	0.3163	0.3163	
R.8 → 16	0.1200	0.3750	0.3750		0.1012	0.5250	0.5250	
R.32 → 16	0.0981	0.3187	0.3187		0.0847	0.4125	0.4125	
G.N20	0.1980	0.3062	0.3363		0.2637	0.5563	0.5250	
G.N30	0.1371	0.3150	0.3212		0.1629	0.5062	0.5000	
M.32	0.1586	0.4700	0.4575		0.1750	0.4750	0.4750	
M.48	0.1301	0.3237	0.3238		0.0953	0.4188	0.4188	
M.128	0.1101	0.4313	0.4313		0.0829	0.3563	0.3563	
M.192	0.0846	0.3000	0.3000		0.1012	0.3500	0.3500	

Table 5: The algorithm efficiency (time/s)

	The proposed algorithm	Without applying spectral subtraction algorithm
Hashing structure	7.24	6.65
Hashing values	0.008	0.008
Total	7.248	6.658

4.3 Robustness for Common Background Noise

People usually talk in a noisy environment, so add Noisex-92 noise library which is common background noise, including white noise, pink noise, factory floor noise 1, factory floor noise 2, speech babble and Volvo noise. The signal-to-noise ratios of noises added are respectively 0db, 10db, 20db, 30db, 40db and 50db.

As shown in Figure 9, this algorithm has extremely strong robustness for Gaussian white noise attack and Babble noise attack; the robustness of this algorithm is obviously stronger than other two algorithm. Its robustness for other several noises is also in the middle level. As for the passing rate for pink noise attack, the passing rate of the three algorithms is all 100%.

As shown in Figure 9, the proposed algorithm has strong robustness for common background noise. In particular, its robustness for Gaussian white noise and Babble noise is significantly higher than the robustness of [4] and [3]. Its robustness performances for Volvo noise, Factory 1 noise, Factory 2 noise and Pink noise are in the middle level. The passing rate of authentication matching for various signal-to-noise ratios is very high. Compared with the NMF algorithm, the tensor decomposition algorithm has stronger robustness for pink noise attack and the feature value decomposed through tensor algorithm has higher stability. Therefore, the algorithm proposed in this paper has strong combination property of robustness for common noises, so it can meet the practical need of speech matching in our daily life.

4.4 Efficiency Analysis

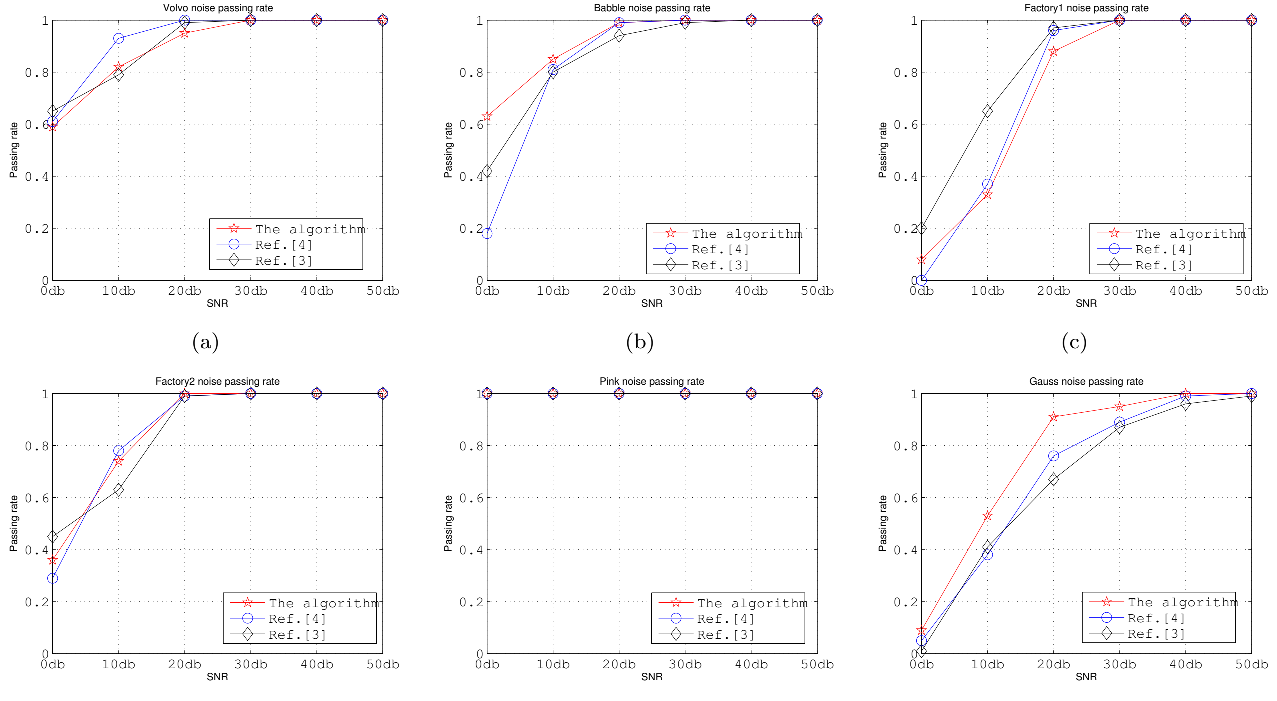
In order to measure the computational efficiency of the proposed algorithm, the researcher randomly extracted 200 segments of speech from the speech library to count the average operating time.

As shown in Table 5, compared with the without applying spectral subtraction algorithm, the proposed algorithm is required to conduct the spectral subtraction noise reduction, wavelet package decomposition prior to feature extraction and make tensor reconstruction for the feature extracted. Thus, the expenditure of operating time is long. On the premise of enhancing the robustness, compared with other algorithms, there is increase in the overall expenditure of operating time of this algorithm and its operating efficiency is largely affected, so this algorithm can only be applied to the occasions with low realtime requirement speech communication.

5 Conclusion

This paper proposed a robust perception hashing speech authentication algorithm based on Spectral subtraction and Multi-feature tensor. Through an experimental discussion and analysis, the proposed algorithm robust is better than the earlier methods as shown in the discussion, the conclusions below can be made.

As shown in the results of speech perception hashing discrimination and collision resistance experiments, the highest speech misidentification probability within the range of threshold. It means that the algorithm has good collision resistance performance and can meet the need of



(a) (b) (c)

(d) (e) (f)

Figure 9: The Speech authentication passing rate of being the common background noise attack. (a)Volvo noise, (b) Babble noise, (c) Factory1 noise, (d) Factory2 noise, (e) Pink noise, (f) Gaussian white noise.

practical application.

As shown in the robust experiment, compared with the other two algorithms, the robustness of the algorithm proposed is improved to a certain extent. After the content preserving operation is conducted, the algorithm can correctly match speech; there is an obvious judging domain in the FAR-FRR curve and the scope of the judging domain is 035. Compared with other algorithm, the FRR-FAR curve has small crossover. The proposed algorithm is especially in allusion to the common background noises during daily communications.

As shown in the experiment of speech against noise attack, this algorithm has strong robustness for common background noise, so it can meet the need of daily communications on varied dialogue backgrounds. Compared with other two kinds of algorithms, the algorithm for common background noise robustness is more stable. This algorithm can control the tensor size as required and model building is flexible. Besides, it can realize the speech content authentication and speaker authentication, the algorithm has a high practical value. Simulations show that the robustness of the proposed algorithm is superior to that without applying spectral subtraction method, but the efficiency is reduced by nearly 1 times and the FAR is increased. The main disadvantage of the proposed algorithm is that the efficiency is deduced. The next of the research objective is to improve the efficiency, decrease the impact of echo and Tamper detection and localization of speech.

Acknowledgment

This study was supported by the Natural Science Foundation of Gansu Province of China under grant 1606RJYA274, the National Nature Science Foundation of China under grant NSFC 61363078. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. T. Ali, J. P. Antoine, J. P. Gazeau, *Coherent States, Wavelets, and Their Generalizations*, New York: Springer Publishing Company, 2013.

[2] J. Chen, S. Xiang, H. Huang and W. Liu, “Detecting and locating digital audio forgeries based on singularity analysis with wavelet packet”, *Multimedia Tools and Application*, vol. 75, no. 4, pp. 2303–2325, 2016.

[3] N. Chen and W. G. Wang, “Robust speech hash function”, *ETRI Journal*, vol. 32, no. 2, pp.345–347, 2010.

[4] N. Chen, H. D. Xiao and W. G. Wan, “Audio hash function based on non-negative matrix factorization of Mel-frequency Cepstral coeffi-cients”, *IET Information Security*, vol. 5, no. 1, pp.7–8, 2013.

[5] M. A. Hossan, S. Memon, M. A Gregory, “A novel approach for MFCC feature extraction”, in *IEEE Xplore Conference: Signal Processing and Communication Systems*, pp.1–5, Gold Coast, Australia, 2010.

[6] Y. B. Huang, Q. Y. Zhang, Z. T. Yuan and Z. P. Yang, "The hash algorithm of speech perception based on the integration of adaptive MFCC and LPCC", *Journal of Huazhong University of Science and Technology*, vol. 43, no. 2, pp.124–128, 2015.

[7] J. Li, L. X. Jin and G. N. Li, "Hyper-spectral remote sensing image compression based on nonnegative tensor factorizations in discrete wavelet domain", *Journal of Electronics & Information Technology*, vol. 35, no. 2, pp. 489–493, 2013.

[8] J. F. Li, T. Wu and H. X. Wang, "Perceptual hashing based on NMF and MDCT coefficient of MFCC for speech authentication", *Journal of Beijing University of Posts and Telecommunications (in Chinese)*, vol. 38, no. 2, pp.89–93, 2015.

[9] P. Lotia and M. R. Khan, "Significance of complementary spectral feature for speaker recognition", *International Journal of Research in Computer and Communication Technology*, vol. 8, no. 2, pp.579–588, 2013.

[10] M. Nouri, N. Farhangian and Z. Zeinolabedini, "Conceptual authentication speech hashing base upon hypotrochoid graph", in *Proceedings of The Sixth International Symposium on Telecommunications*, pp. 1136–1141, Tehran, Iran, Nov. 2012.

[11] H. Ozer, B. Sankur and N. Memon, "Perceptual audio hashing functions", *EURASIP Journal on Applied Signal Processing*, vol. 12, pp. 1780–1793, 2005.

[12] V. Panagiotou and N. Mitianoudis, "PCA summarization for audio song identification using Gaussian mixture models", in *Proceedings of The 18th International Conference on Digital Signal Processing*, pp.1–6, Santorini, Greece, July 2013.

[13] J. W. Picone, "Signal modeling techniques in speech recognition", *Proceedings of the IEEE*, vol. 81, no. 9, pp. 1215–1247, 1993.

[14] M. Ramona and G. Peeters, "Audio identification based on spectral modeling of bark-bands energy and synchronization through onset detection", in *Proceedings of 2011 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP'11)*, pp.477–480, Prague, Czech, May 2011.

[15] P. Sharma, K. Khan and K. Ahmad, "Image denoising using local contrast and adaptive mean in wavelet transform domain", *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 12, no. 6, pp.1450038.1–1450038.15, 2014.

[16] R. Sharma and V. P. Pyara, "A robust denoising algorithm for sounds of musical instruments using wavelet packet transform", *Circuits and Systems*, vol. 7, no. 4, pp. 459, 2013.

[17] B. Q. Xu, Q. Xiao, Z. X. Qian, and C. Qin, "Unequal protection mechanism for digital speech transmission based on turbo codes", *International Journal of Network Security*, vol. 17, no. 1, pp. 85–93, 2015.

[18] Y. Yang and S. Nagarajaiah, "Blind identification of damage in time-varying systems using independent component analysis with wavelet transform", *Mechanical Systems and Signal Processing*, vol. 47, no. 1, pp. 3–20, 2014.

[19] Q. Y. Zhang, W. J. Hu, S. B. Qiao and Y. B. Huang, "Speech perceptual hashing authentica-tion algorithm based on spectral subtraction and energy to entropy ratio", *International Journal of Network Security*, vol. 19, no. 5, pp.752–760, 2017.

[20] Q. Y. Zhang, P. F. Xing, Y. B. Huang, R. H. Dong and Z. P. Yang, "An efficient speech perceptual hashing authentication algorithm based on wavelet packet decomposition", *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 2, pp. 311–322, 2015.

[21] Q. Y. Zhang, Z. P. Yang, Y. B. Huang, R. H. Dong and P. F. Xing, "Efficient robust speech authentication algorithm for perceptual hashing based on Hilbert-Huang transform", *Journal of Information and Computational Science*, vol. 11, no. 18, pp. 6537–6547, 2014.

[22] Q. Y. Zhang, Z. P. Yang, Y. B. Huang, S. Yu and Z. W. Ren, "Robust speech perceptual hashing algorithm based on linear predication residual of G.729 speech codec", *International Journal of Innovative, Computing, Information and Control*, vol. 11, no. 6, pp.2159–2175, 2015.

[23] Y. Zhang and Y. Zhao, "Real and imaginary modulation spectral subtraction for speech enhancement", *Speech Communication*, vol. 55, no. 4, pp.509–522, 2013.

Biography

Yi-bo Huang received Ph.D candidate degree form Lanzhou university of technology in 2015, and now working as a lecturer in the college of physics and electronic engineering in northwest normal university, He main research interests include Multimedia in-formation processing, information security, speech recognition.

Qiu-yu Zhang (Researcher/Ph.D supervisor), graduated from Gansu university of technology in 1986,and then worked at school of computer and communication in Lanzhou university of technology. He is vice dean of Gansu manufacturing information engineering research center, CCF senior member, a member of IEEE and ACM. His research interests include network and information security, information hiding and steganalysis, multimedia communication technology.

Cryptanalysis of Design and Analysis of a Provably Secure Multi-server Authentication Scheme

Naresh Babu Muthu Mohan[1], Ardhani Sathya Narayana Chakravarthy[1], Cherukuri Ravindranath[2]

(Corresponding author: Naresh Babu Muthu Mohan)

Department of Computer Science and Engineers, Jawaharlal Nehru Technological University[1]
Jawaharlal Nehru Technological University Road, Kakinada, Andhra Pradesh 533003, India
Trinity Institute Of Technology & Research[2]
Kokta By-Pass Road, Near Hindustan Petrol Pump,, Raisen Rd, Bhopal, Madhya Pradesh 462021, India
(Email:itsnaresh4u@gmail.com[1], asnchakravarthy@yahoo.com[2], ravindranathc@gmail.com[3])

(Received Mar. 29, 2016; Revised and Accepted June 10, 2016)

Abstract

The rapid growth of inter-networking and communication technologies resulted in an exponential hit rate on commercial service providing websites (servers) like Google, Amazon, Flipkart etc. from remote users connected via Internet. To handle the networking load, the organizations are moving from the traditional two tier client server architecture to multi-server architecture for efficient load balancing. The traditional two-party authentication protocol for remote user authentication are not sufficient to break the ever increasing attacks on open network i.e. Internet. Also, the existing two-party authentication protocols are meant for single server, adopting these protocols for multi-server environment results in the requirement of huge computation cost for separate registration of user at each server. So, researchers started proposing authentication schemes specific to multi-server environment. In 2014, Yeh et al. proposed an improved version over Pippal et al.'s scheme which eliminates all identified weaknesses like susceptible to user impersonation attack, server counterfeit attack, and the man-in-the-middle attack. In 2015, Mishra et al. demonstrated that Yeh et al. scheme is susceptible to off-line password guessing attack, insider attack and user impersonation attack and proposed an improved version. In this manuscript we do a thorough analysis on Mishra et al. scheme and determine that Mishra et al. scheme is liable to 'known session specific temporary information' attack and based on that, the attacker can realize all key attacks. We also demonstrate that Mishra et al. scheme consists of major inconsistencies like 'inefficient login phase' which restrict the protocol to adopt to real time implementation.

Keywords: Authentication; Elliptic Curve Cryptography; Multi-server Authentication; Smart Cards

1 Introduction

The advances in internet, mobile and networking technologies resulted in an exponential access to remote servers using high end mobile devices on the go via Internet (as shown in Figure 1). The traditional authentication schemes are primarily proposed keeping in mind the traditional two-tier client-server architecture and traditional communicating devices like desktop etc. [2, 4, 8, 15, 16]. Due to advances in mobile and communication technologies, users are able to connect to remote servers through mobile devices on the go, which results in an increased hit rate on e-commerce servers. Hence, all the small and medium enterprises are moving to a multi-server environment [3, 7, 14]. Due to this, there is a critical need for robust, efficient and lightweight remote user authentication algorithms. On the one hand, adopting these protocols for multi-server environment results in the users need to register in each server and to store large sets of data, including identities and passwords [1, 5, 17, 26].

Various researchers had proposed authentication protocols for secure authentication of users connecting to remote servers based on various techniques like usage of verification table [13], symmetric key cryptosystem [10], dynamic Identity based [6, 9, 19, 22, 23, 24, 25], modified password based [23, 24], involvement of the registration center in the authentication process [9].etc. Unfortunately, most of the protocols are analyzed insecure shortly, after they were put forward [6, 9, 10, 12, 13, 21, 22, 24]. Meanwhile, identity protection is considered to be important for authentication and key agreement protocol design in single-server and multi-server architectures.

In 2013, Pippal et al. [21] proposed a robust multi-server authentication scheme based on smart cards, with added advantages like elimination of verifier table, registered remote users are allowed to access multiple servers

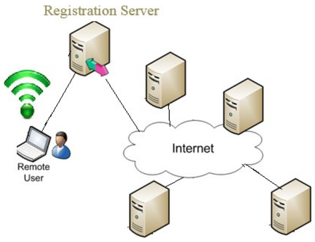


Figure 1: Typical multi server environment

without multiple registration. Also the registered users can alter the password securely without any assistance from the registration center or remote server. In 2014, Yeh et al. [25] demonstrated that the remote user multi server authentication scheme proposed by Pippal et al. [21] is vulnerable to user impersonation attack, server counterfeit attack, and the man-in-the-middle attack and having proved the inconsistencies in Pippal et al. [21] scheme, Yeh et al. proposed an improved version, which eliminates all identified weaknesses with the same order of computation complexity.

In 2015, Mishra et al. [19] did a thorough literature analysis of multi-server authentication schemes and summarized that most of the existing multi-server authentication schemes require all the involved servers to be trusted, involvement of registration center or central authority in mutual authentication [20] or multiple secret keys. In practical scenarios, the servers may be semi-trusted, thus considering all servers to be trusted does not seem to be realistic scenario. Involvement of registration center/central authority in the computation process like mutual authentication may create a bottleneck scenario for a large network, which is a draw back in multi-server authentication scheme proposed by odelu et al. [20]. Also, computation of multiple secret keys may not be suitable for smart card based environment as smart card keeps limited storage space. In sound literature analysis, Mishra et al. [19] demonstrated that recently proposed Yeh et al. [25] multi-server authentication scheme is susceptible to off-line password guessing attack, insider attack and user impersonation attack. Having found the security pitfalls, Mishra et al. [19] proposed an improved multi-server authentication scheme which does not require all servers to be trusted, central authority no longer needed in authentication and smart card need not to store multiple secret keys.

On thorough analysis of Mishra et al. [19] multi-server authentication scheme, we demonstrate that their scheme is susceptible to session specific temporary information attack, on the success of it, Mishra et al. [19] scheme is susceptible to leakage of user identity, password and computation of session key by the attacker. We also established that Mishra et al. [19] scheme includes major inconsistencies in which lack of early detection of wrong credentials by the smart card, which results in excessive computation on the server side, which ultimately results in a Denial of Service attack. In future work, we aim to propose a secure and light weight multi server authentication scheme by eliminating the security pitfalls and inconsistencies found in Mishra and other related schemes.

The rest of the paper is organized as follows. In Section 2, a brief review of Mishra et al. scheme is given. Section 3, describes the security weakness of Mishra et al. scheme. Section 4 provides the conclusion of the paper.

2 Review of Mishra et al. Scheme

In this section, we examine Design and Analysis of a Provably Secure Multi-server Authentication Scheme by Mishra et al. [19] in 2015 and then demonstrate its security pitfalls. The notations used in Mishra et al. [19] are listed in Table 1.

Table 1: The notations used in Mishra et al. [19]

Parameter	***Description***
U_i	$User_i$
R.S	A trustworthy Registration center / Registration server
S_j	j^{th} server in the system
UID_i	Unique identity of $User_i$
UPW_i	Unique password of $User_i$
T_i	Timestamp generated by entity i
SK_{ij}	Session key between $User_i$ and $server_j$
MK	Master key of RS
USK_i, UPK_i	i^{th} user secret/public key
h(.), h1(.), h2(.)	One-way hash functions
p	A large prime number
$E_p(a,b)$	An elliptic curve $y^2 = x^3 + ax + b(mod)p$ over a finite Z_p with $4a^3 + 27b^2 \neq 0((modp))$ based on group G
$\oplus$	Bitwise XOR operation
$\|\|$	Bitwise string concatenation

2.1 Registration Server (R.S)

An additive group G, whose generator is P. G is a set of points over an elliptic curve EP(a,b) of order n.

Select:

$$\begin{aligned} h: &\quad \{0,1\}^* \rightarrow \{0,1\}^k, \\ h1: &\quad \{0,1\}^* * G \rightarrow \{0,1\}^*, \\ h2: &\quad \{0,1\}^* * \{0,1\}^* * \{0,1\}^* \ * G * G * G \rightarrow \{0,1\}^k \end{aligned}$$

Chooses: A master key MK of 1024 bits.

Registration server makes as public $\{EP(a,b)$, P, $h(.)$, $h1(.)$, $h2(.)\}$ and keeps its master key MK as private.

Figure 2: Server registration phase of Mishra scheme

Server (S_j) ***Registration Server (R.S)***

Selects: Identity SID_j

$\{SID_j\}$

$- - - - - - - \rightarrow$

Computes: $SSK_j = h(SID_j||MK)$

$SPK_j = SSK_j.P$

$\{SSK_j, SPK_j\}$

$\leftarrow - - - - - - -$

Figure 3: User registration phase of Mishra scheme

User (U_i) ***Registration Server*** $(R.S)$

Selects: UID_i, PW_i, N

$RPW_i = h(PW_i||UID_i)$.

$\{RPW_i \oplus N, UID_i\}$

$- - - - - - - - - - - \rightarrow$

$USK_i = h(UID_j||MK)$

$X_i = USK_i \oplus RPW_i \oplus N$

$UPK_i = USK_i.P$

R.S $\rightarrow$ (UID_i, UPK_i) to all servers

S.C $= \{X_i, P, h(.), h1(.), h2(.)\}$

$\leftarrow - - - - - - - - - - - - - -$

$Y_i = X_i \oplus N = USK_i \oplus RPW_i$

Replaces $X_i with Y_i$ in S.C.

i.e. S.C $= \{Y_i, P, h(.), h1(.), h2(.)\}$

The registration server(RS) performs the following steps in offline mode before the actual deployment of the servers in deployment field.

Step 1. R.S selects a large odd prime number 'p' of minimum 160 bits, generates a Galois Field G.F (p) and elliptic curve Ep(a,b), which is a set of all points on the curve $y^2 = x^3 + ax + b(modp)$, such that a,b $\epsilon Z_p = \{0, 1, 2, 3....p-1\}$, satisfying the condition $4a^3 + 27b^2 \neq 0$. 'G' represents the base point of elliptic curve 'E' of order 'n', which is of 160 bits such that $n > \sqrt{p}$. R.S chooses three hash functions h, $h1$, $h2$ and opts MK as its master key.

Step 2. Registration server makes as public $\{EP(a, b)$, P, $h(.)$, $h1(.)$, $h2(.)\}$ and keeps its master key MK as private.

2.2 Server Registration Phase

This phase is invoked whenever a server S_j registers with the registration server for the first time. The registration server assigns secret and public keys to the server.

This phase is invoked whenever a server S_j registers with the registration server for the first time (Figure 2).

Step 1. The server S_j selects the identifier SID_j and provides its identity $\{SID_j\}$ to the registration server via a secure channel for registration.

Step 2. On receiving the registration request $\{SID_j\}$, RS computes the secret and public keys for S_j as follows:

$$SSK_j = h(SID_j||MK), SPK_j = SSK_j.P$$

where MK is its secret master key. R.S submits $\{SSK_j, SPK_j\}$ to S_j, through a secure communication channel.

2.3 User Registration Phase

This phase is invoked whenever a user U_i registers with the registration server for the first time (Figure 3).

Step 1. The user U_i selects the identifier UID_i, a random number N, and the password PW_i. U_i then computes $RPW_i = h(PW_i||UID_i)$. U_i submits the registration request $\{RPW_i \oplus N, UID_i\}$ to the registration server via a secure channel for registration.

Step 2. On receiving the login request $\{RPW_i \oplus N, UID_i\}$, the RS performs the following computations to compute the secret and public key for U_i. $USK_i = h(UID_j||MK)$, $UPK_i = USK_i.P, X_i = USK_i \oplus RPW_i \oplus N$. R.S forward the secret and public key pair (USK_i, UPK_i) of U_i to all registered servers. Finally, the RS issues a tamper-proof smart card with the following parameters stored in it $S.C = X_i, P, h(.), h1(.), h2(.)$ to U_i through a secure communication channel.

Figure 4: Login phase of Mishra scheme

$\mathbf{U_i/S.C(UPK_i, USK_i, SPK_j)}$	$\mathbf{Server(S_j)(UPK_i, SSK_j, SPK_j)}$
Submits UID_i, PW_i.	
Computes: $RPW_i = h(PW_i \vert\vert UID_i)$	
Retrieves $USK_i = Y_i \oplus RPW_i$.	
Selects the targeted server S_j to access the resources.	
Achieves the S_j public key from R.S public directory i.e. (SID_j, SPS_j)	
Generates a session specific arbitrary number $'r_u'$	
Computes: $A_i = r_u.P$	
$B_{ij} = USK_i.SPK_j (= USK_i.SSK_j.P = UPK_i.SSK_j = USK_i.SSK_j.P)$.	
$C_{ij} = r_u.SPK_j = (r_u.SSK_j.P = r_u.P.SSK_j = A_i.SSK_j)$	
$V_i = h(UID_i \vert\vert SID_j \vert\vert T1 \vert\vert B_{ij} \vert\vert C_{ij} \vert\vert A_i)$	
$DID_i = UID_i \oplus h1(SID_j \vert\vert C_{ij})$	
$\{DID_i, A_i, V_i, T1\}$ $- - - - - - - - - - - \rightarrow$	

Step 3. On receiving S.C from R.S, U_i computes $Y_i = X_i \oplus N = USK_i \oplus RPW_i$ and replaces X_i with Y_i in its S.C.

Finally the U_i S.C contains the parameters: $\{Y_i$, P, $h(.)$, $h1(.)$, $h2(.)\}$.

2.4 Login Phase

Whenever the user U_i wants to access data from a server S_j deployed in a multi-server environment, the user U_i needs to perform the following steps (Figure 4).

Step 1. U_i inserts his/her smart card into the card reader of a specific terminal and provides his/her Identity UID_i, password PW_i.

Step 2. The S.C computes $RPW_i = h(PW_i||UID_i)$ and retrieves U_i secret key $USK_i = Y_i \oplus RPW_i$.

Step 3. The S.C achieves the server S_j public key from the R.S public directory, i.e (SID_j, SPS_j). S.C generates a session specific arbitrary number $'r_u'$.

Step 4. The smart card then computes the variables $A_i = r_u.P$, $B_{ij} = USK_i.SPK_j (= USK_i.SSK_j.P = UPK_i.SSK_j = USK_i.SSK_j.P)$. $C_{ij} = r_u.SPK_j = (r_u.SSK_j.P = r_u.P.SSK_j = A_i.SSK_j)$,
$V_i = h(UID_i||SID_j||T1||B_{ij}||C_{ij}||A_i)$, Masked identity $DID_i = UID_i \oplus h1(SID_j||C_{ij})$ where T1 is the current time stamp.

Step 5. The S.C finally forwards the login message $\{DID_i, A_i, V_i, T1\}$ to RS, via a public channel.

2.5 Authentication Phase

On receiving the login request $\{DID_i, A_i, V_i, T1\}$ at time, from S.C, at time $T1^*$, the server S_j validates the login request by checking whether $(T1^* - T1) \leq \Delta t$, then S_j proceeds as follows (Figure 5).

Step 1. Compute $A_i.SSK_j = r_u.P.SSK_j = r_u.SPK_j = C_{ij}{}^*$. A_i is received through login request by U_i. Retrieve $UID_i{}^* = DID_i \oplus h1(SID_j||C_{ij}{}^*)$.

Step 2. Compute: $B_{ij}{}^* = UPK_i.SSK_j$, $V_i^* = h(UID_i||SID_j||T1||B_{ij}{}^*||C_{ij}{}^*||A_i)$. Validate whether $V_i = V_i^*$ if yes, U_i is authenticated.

Step 3. Generates a session specific arbitrary number $'r_s'$, Compute: $D_j = r_s.P$, $E_j = r_s.A_i = r_s.r_u.P.$. $SK_{ij} = h(UID_i||SID_j||T1||B_{ij}{}^*||C_{ij}{}^*||E_j)$, $V_j = h(UID_i||SK_{ij}||T2||B_{ij}{}^*||C_{ij}{}^*||D_j)$ And forwards the login reply message $\{D_j, V_j, T2\}$ to S.C via a public channel.

Step 4. On receiving the login reply message, S.C computes:

$$\begin{aligned} E_j{}^* &= r_u.D_j = r_u.r_s.P, \\ SK_{ij}{}^* &= h(UID_i||SID_j||T1||B_{ij}||C_{ij}||E_j{}^*), \\ V_j{}^* &= h(UID_i||SK_{ij}{}^*||T2||B_{ij}||C_{ij}||D_j). \end{aligned}$$

Validate whether $V_j = V_j{}^*$, if yes, S_j is authenticated. If yes, S.C authenticates the server S_j.

3 Cryptanalysis of Mishra et al. Scheme

In this segment, we will cryptanalyze the Mishra et al. [19] scheme and illustrate that Mishra et al. scheme is vulnerable to Known Session Specific Temporary Information Attack, i.e. if session specific arbitrary numbers, i.e. r_u and r_s are leaked out, then an attacker can achieve the secret key of U_i, password PW_i of U_i, session key SK_{ij}. We describe the detailed steps in attack as follows:

Figure 5: Authentication phase of Mishra scheme

$\mathbf{U_i/S.C}$ $(\mathbf{UPK_i, USK_i, SPK_j})$		$\mathbf{Server(S_j)(UPK_i, SSK_j, SPK_j)}$
	$\{DID_i, A_i, V_i, T1\}$ $- - - - - - - \rightarrow$	
		Receive at $T1^*$. Check $(T1^* - T1) \leq \Delta t$ Compute: $A_i.SSK_j = r_u.P.SSK_j = r_u.SPK_j = C_{ij}{}^*$ // A_i is received through login request by U_i. Retrieve: $UID_i{}^* = DID_i \oplus h1(SID_j \vert\vert C_{ij}{}^*)$ $B_{ij}{}^* = UPK_i.SSK_j$ $V_i{}^* = h(UID_i \vert\vert SID_j \vert\vert T1 \vert\vert B_{ij}{}^* \vert\vert C_{ij}{}^* \vert\vert A_i)$ Validate whether $V_i = V_i{}^*$ if yes, U_i is authenticated. Genrates a session specific arbitrary number $'r'_s$ Compute: $D_j = r_s.P$, $E_j = r_s.A_i = r_s.r_u.P$. $SK_{ij} = h(UID_i \vert\vert SID_j \vert\vert T1 \vert\vert B_{ij}{}^* \vert\vert C_{ij}{}^* \vert\vert E_j)$ $V_j = h(UID_i \vert\vert SK_{ij} \vert\vert T2 \vert\vert B_{ij}{}^* \vert\vert C_{ij}{}^* \vert\vert D_j)$
	$\{D_j, V_j, T2\}$ $\leftarrow - - - - - - -$	
Receive at $T2^*$ Check: $(T2^* - T2) \leq \Delta t$ $E_j{}^* = r_u.D_j = r_u.r_s.P$ $SK_{ij}{}^* = h(UID_i \vert\vert SID_j \vert\vert T1 \vert\vert B_{ij} \vert\vert C_{ij} \vert\vert E_j{}^*)$ $V_j{}^* = h(UID_i \vert\vert SK_{ij}{}^* \vert\vert T2 \vert\vert B_{ij} \vert\vert C_{ij} \vert\vert D_j)$ Validate whether $V_j = V_j{}^*$ if yes, S_j is authenticated.		

1) An opponent or an attacker or legal user can extract the information cached in the smart card by several techniques such as power consumption or leaked information [11, 18], etc. i.e. $S.C = \{Y_i, P, h(.), h1(.), h2(.)\}$.

2) An opponent can passive monitor or eavesdrop or alter or replay the login request, login reply messages communicated among U_i and R.S over a public channel which is Internet, i.e. $\{\{DID_i$, A_i, V_i, $T1\}$, $\{D_j$, V_j, $T2\}\}$.

An attacker is supposed to have access to all the values discussed in Table 2, based on these the attacker can accomplish various attacks as discussed below.

3.1 Fails to Resist Known Session Specific Temporary Information Attack

The compromise of session specific arbitrary numbers should not allow the attacker to compute any unknown value of the communication participants and should not compromise the computed the session key.

Case 1: Offline Identity computation by an attacker.
In Mishra et al. scheme assume that the session specific arbitrary number r_u, r_s are compromised and an attacker got hold of it. As discussed above, the attacker is having access to the values as discussed in Table 2, can perform following steps:

Step 1: Compute $C_{ij}{}^* = r_u.SPK_j.SPK_j$ is a server public key which is known to all the participants.

Step 2: From the intercepted login message $\{DID_i, A_i, V_i, T1\}$, retrieve UID_i from DID_i using $C_{ij}{}^*$ computed in Step 1, i.e. $UID_i = DID_i \oplus h1(SID_j \mid\mid C_{ij}{}^*)$. Hence Mishra et al. scheme failed to preserve user anonymity.

Case 2: Offline Password guessing by an attacker.

Step 1: 'E' can retrieve Y_i from the U_i S.C and can frame:

$$\begin{aligned} Y_i &= X_i \oplus N = USK_i \oplus RPW_i \\ &= USK_i \oplus h(PW_i \vert\vert UID_i). \\ USK_i &= Y_i \oplus h(PW_i \vert\vert UID_i). \qquad (1) \end{aligned}$$

Step 2: Compute

$$C_{ij} = r_u.SPK_j. \qquad (2)$$

We are assuming r_u is compromised and SPK_j is the server public key.

Step 3: Replace Equation (1) in place of USK_i.

$$\begin{aligned} B_{ij} &= USK_i.SPK_j \\ &= (Y_i \oplus h(PW_i \vert\vert UID_i)).SPK_j. \qquad (3) \end{aligned}$$

On intercepting $V_i, A_i, T1$ from the login request message sent by U_i to S_j, the attacker 'E' can

Table 2: Values known and unknown to an attacker

Values Known to the Attacker	Values Known to the Attacker	Values doesn't known to the Attacker
A legal adversary 'E' is assumed to know: 1. The smart card values of legal user U_i.	1. $\{Y_i, P, h(.), h1(.), h2(.)\}$ $Y_i = USK_i \oplus RPW_i$	1. SSK_j, UID_i, USK_i, MK
2. The intermediate communication messages exchanged between U_i and S.	2. $\{\{DID_i,\ A_i,\ V_i,\ T1\}, \{D_j,\ V_j,\ T2\}\}$	
3. All public values of U_i and S_j	3. SPK_j, SID_j, UPK_j	

proceed as follows to compute the U_i password from Equation (3).

Step 3.1: $V_i = h(UID_i \,||\, SID_j \,||\, T1 \,||\, B_{ij} \,||\, C_{ij} \,||\, A_i)$. In V_i, 'E' knows UID_i, SID_j, $T1$, A_i, C_{ij} and B_{ij} are computed in step 2 and Step 3 above.

Step 3.2: Substitute B_{ij} in V_i. i.e $V_i = h(UID_i \,||\, SID_j \,||\, T1 \,||\, B_{ij} \,||\, C_{ij} \,||\, A_i) = h(UID_i \,||\, SID_j \,||\, T1 \,||\, (Y_i \oplus h(PW_i \,||\, UID_i)).\ SPK_j \,||\, C_{ij} \,||A_i)$ using Equation (3).

Step 3.3: In V_i of Step 3.2, the only unknown parameter to an attacker is PW_i. As discussed in [6, 12, 23], if in Mishra et al. [19] scheme, if the user U_i opts for a password, which is a weak (low entropy), the attacker can perform password guessing attack as follows similar to [6, 24]:

Step 3.3.1: Guesses the value of PW_i to be $PW_i{}^*$ from a dictionary space ∂.

Step 3.3.2: Compute: $V_i^* = h(UID_i||SID_j||T1||(Y_i \oplus h(PW_i{}^*||UID_i)))$. Check computed V_i* equal to V_i in the intercepted login request. If yes, the U_i original password is $PW_i{}^*$ else 'E' proceeds to Step 3.3.1.

Hence, as discussed above in Mishra et al. scheme, the attacker succeeds to guess the low-entropy password PW_i.

Step 4: On getting the password PW_i of U_i, the attacker 'E' can compute the U_i secret key as follows:

$$\begin{aligned} Y_i &= X_i \oplus N = USK_i \oplus RPW_i \\ &= USK_i \oplus h(PW_i||UID_i). \\ USK_i &= Y_i \oplus h(PW_i||UID_i). \end{aligned}$$

As 'E' knows Y_i from U_i smart card and UID_i as discussed in Case 1.

Hence, we can confirm that Mishra et al. suffers from the biggest drawback that, on compromise f session specific arbitrary numbers, all the secret parameters of protocol participants can be find out.

Case 3: Computing the session key by an attacker.

In Mishra et al. [19] scheme, the session key $SK_{ij} = h(UID_i||SID_j||T1||B_{ij}{}^*||C_{ij}{}^*||E_j)$. As discussed above, in SK_{ij}, 'E' knows all the values except E_j. As discussed above, if r_u, r_s are compromised, 'E' can compute $E_j = r_u.r_s.P$. Hence, based on above discussion, we can confirm that in Mishra et al. scheme, if r_u, r_s are compromised, the attacker 'E' can compute U_i identity i.e. UID_i, password PW_i and session key SK_{ij}.

3.2 Fails to Resists Denial of Service Attack (Inefficient Login Phase)

Assume that the legal user provides a wrong password PW_R. Instead of PW_i during login stage.

Mishra et al. [19] scheme is not secured against computation exhaustive attacks like denial of service attack as there is no verification of user data by S.C during the login phase. Thus if a legal user U_i submits a wrong password PW_r instead of PW_i, as discussed in [6, 23], SC performs all calculations to compute the login request without verifying the correctness of inserted identity ID and password PW. This loophole endangers the security of the scheme in following ways (Figure 6).

Offline and online password guessing attack, user impersonation attack, and denial of service attack. Network Flooding with wrong login request above, the smart card still proceeds further to compute the login message which is a fake login request messages to the server which leads to the excessive computation on the server side. Similarly to guess the password correctly, an adversary sends the guessed password online a number of times till she will not succeed which leads to excessive computation on server as smart card lacks any verification mechanism. Thus proto-

Figure 6: Denial of service attack

$\mathbf{U_i/S.C\ (UPK_i, USK_i, SPK_j)}$ $\mathbf{Server(S_j)(UPK_i, SSK_j, SPK_j)}$

U_i provides UID_i, PW_R.
PW_R is a wrong password instead of his correct password PW_i
Computes: $RPW_R = h(PW_R||UID_i)$
Retrieves $USK_R = Y_i \oplus RPW_R$.
Selects the targeted server S_j to access the resources.
Achieves the S_j public key from R.S public directory i.e. (SID_j, SPS_j)
Generates a session specific arbitrary number $'r'_u$
Computes: $A_i = r_u.P$
$B_{R_{ij}} = USK_R.SPK_j$
$C_{ij} = r_u.SPK_j$
$V_{R_i} = h(UID_i||SID_j||T1||BR_{ij}||C_{ij}||A_i)$
$DID_i = UID_i \oplus h1(SID_j||C_{ij})$

$\{DID_i, A_i, V_i, T1\}$
$------\rightarrow$

Compute: $A_i.SSK_j = r_u.P.SSK_j = r_u.SPK_j = C_{ij}*$
// A_i is received through login request by U_i.
Retrieve: $UID_i* = DID_i \oplus h1(SID_j||C_{ij}{}^*)$
$B_{ij}{}^* = UPK_i.SSK_j$
$V_i{}^* = h(UID_i||SID_j||T1||B_{ij}{}^*||C_{ij}{}^*||A_i)$
Verifies $V_i{}^*$ equal V_i
As the verification fails, R.C rejects the message.

col is not secure against denial of service attack. Due to inefficient login phase, it costs, 3Hash operations +3Elliptic Point Multiplication operations.

4 Conclusion

Recently Mishra et al. proposed an ECC-based multi-server authentication scheme. Even though it is a novel attempt, after thorough analysis of Mishra et al. paper, we demonstrated that their scheme is vulnerable to known session specific temporary information attack which results in leakage of user identity, password and computation of session key by the attacker. We also established that Mishra et al. scheme include major inconsistencies in which lack of early detection of wrong credentials by S.C, which results in excessive computation on the server side, which ultimately results in Denial of Service attack. In future work, we aim to propose a secure and light weight multi server authentication scheme by eliminating the security pitfalls and inconsistencies found in Mishra and other related schemes.

References

[1] R. Amin, “Cryptanalysis and efficient dynamic id based remote user authentication scheme in multi-server environment using smart card,” *International Journal of Network Security*, vol. 18, no. 1, pp. 172–181, 2016.

[2] N. Anwar, I. Riadi, A. Luthfi, “Forensic SIM card cloning using authentication algorithm,” *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 71–81, 2016.

[3] C. C. Chang, W. Y. Hsueh, T. F. Cheng, “An advanced anonymous and biometrics-based multi-server authentication scheme using smart cards,” *International Journal of Network Security*, vol. 18, no. 6, pp. 1010–1021, 2016.

[4] T. Y. Chang, W. P. Yang, M. S. Hwang, ”Simple authenticated key agreement and protected password change protocol”, *Computers & Mathematics with Applications*, vol. 49, pp. 703–714, 2005.

[5] T. Y. Chen, C. C. Lee, M. S. Hwang, J. K. Jan, ”Towards secure and efficient user authentication scheme using smart card for multi-server environments,” *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, Nov. 2013.

[6] A. K. Das, “A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks”, *Peer-to Peer Networking and Applications*, 2014.

[7] T. H. Feng, C. H. Ling, and M. S. Hwang, ”Cryptanalysis of Tan’s Improvement on a Password Authentication Scheme for Multi-server Environments,”

International Journal of Network Security, vol. 16, no. 4, pp. 318–321, 2014.

[8] M. S. Hwang, C. C. Lee, Y. L. Tang, ”An improvement of SPLICE/AS in WIDE against guessing attack”, *International Journal of Informatica*, vol. 12, no. 2, pp.297-302, Apr. 2001.

[9] D. He and D. Wang, “Robust biometrics-based authentication scheme for multiserver environment”, *IEEE System Journal*, vol. 9, no. 3, pp. 816–823, 2015.

[10] W. S. Juang, “Efficient multi-server password authenticated key agreement using smart cards”, *IEEE Transactions on Consumer Electronics*, vol. 12, no. 6, pp. 251–255, 2004.

[11] P. Kocher, J. Jaffe and B. Jun, “Differential power analysis”, in *Advances in Cryptology*, pp. 388–397, Springer, 1999.

[12] S. Kumari, M. K. Khan and R. Kumar, “Cryptanalysis and improvement of a privacy enhancedscheme for telecare medical information systems”, *Journal of Medical Systems*, vol. 37, pp. 37, 2013.

[13] L. H. Li, L. C. Lin and M. S. Hwang, “A remote password authentication scheme for multiserver architecture using neural networks”, *IEEE Transactions on Neural Networks*, vol. 50, no. 1, pp. 1498–1504, 2001.

[14] I. C. Lin, M. S. Hwang, L. H. Li, ”A new remote user authentication scheme for multi-server architecture”, *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.

[15] C. W. Lin, C. S. Tsai, M. S. Hwang, “A new strong-password authentication scheme using one-way hash functions”, *International Journal of Computer and Systems Sciences*, vol. 45, no. 4, pp. 623–626, July 2006.

[16] Y. Liu, C. C. Chang and S. C. Chang, “An efficient and secure smart card based password authentication scheme,” *International Journal of Network Security*, vol. 19, no. 1, pp. 1–10, 2017.

[17] Y. Liu, C. C. Chang, C. Y. Sun, “Notes on ’An anonymous multi-server authenticated key agreement scheme based on trust computing using smart card and biometrics’,” *International Journal of Network Security*, vol. 18, no. 5, pp. 997–1000, 2016.

[18] T. S. Messerges, E. A. Dabbish, R. H. Sloan, “Examining smart-card security under the threat of power analysis attacks”, *IEEE Transactions on Computers*, vol. 51, pp. 541–552, 2002.

[19] D. Mishra, “Design and analysis of a provably secure multi-server authentication scheme”, *Wireless Personal Communications*, vol. 86, pp. 1095–1119, 2016.

[20] V. Odelu, A. K. Das and A. Goswam, “A secure biometrics-based multi-server authentication protocol using smart cards”, *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[21] R. S. Pippal, C. D. Jaidhar, and S. Tapasw, “Robust smart card authentication scheme for multi-server architecture”, *Wireless Personal Communications*, vol. 72, pp. 72, 2013.

[22] J. L. Tsai, “Efficient multi-server authentication scheme based on one-way hash function withoutverification table”, *Computers and Security*, vol. 27, no. 3, pp. 115–121, 2008.

[23] J. L. Tsai and N. W. Lo, “A new password-based multi-server authentication scheme robust to password guessing attacks”, *Wireless Personal Communications*, vol. 71, pp. 1977–1988, 2013.

[24] B. Wang and M. Ma, “A smart card based efficient and secured multi-server authentication scheme”, *Journal of Wireless Personal Communication*, vol. 68, no. 2, pp. 361–378, 2013.

[25] K. H. Yeh, “A provably secure multi-server based authentication scheme”, *Wireless Personal Communications*, vol. 79, pp. 1621–1634, 2014.

[26] H. Zhu, Y. Zhang, and Y. Sun, “Provably secure multi-server privacy-protection system based on chebyshev chaotic maps without using symmetric cryptography,” *International Journal of Network Security*, vol. 18, no. 5, pp. 803–815, 2016.

Biography

Naresh Babu Muthu Mohan received his M.Tech degree from VIT University, Vellore, India. Currently he is pursuing Ph. D. in the Department of C.S.E., Jawaharlal Nehru Technological University Kakinada, Kakinada, A.P., India. He has published papers in various International journals and conferences. His areas of current research include Networks, Mobile Security & Cryptography.

Ardhani Sathya Narayana Chakravarthy is currently working as Professor, Dept. of Computer Science & Engineering, Coordinator MOOCs & Skill Development Centre, Jawaharlal Nehru Technological University Kakinada,Kakinada, A.P., India. He has 62 papers published in various International journals and conferences. His research areas include Networks, Security & Cryptography, Biometrics, and Digital Forensics. He is Editorial board member for various International Journals.

Cherukuri Ravindranath received the Ph.D degree in Electrical and Computer Engineering from University of Texas at San Antonio(USA). He is currently the principal of Trinity Institute of technology & Research, Bhopal, India. He is reviewer of IEEE, SPIE, Elsevier. He was post research fellow at University of Texas at San Antonio(USA). He has published large number of research papers in the reputed International/National journals and conference proceedings. He has two patents in Digital Image Security. His areas of current research interest include System Security, Multimedia Processing, Information Assurance and Applied Statics.

Defeating Cyber Attacks Due to Script Injection

Debasish Das, Dhruba Kumar Bhattacharyya
(Corresponding author: Debasish Das)

The Department of Computer Science and Engineering, Tezpur University[1]
Napaam, Sonitpur, Assam 784028, India
(Email: ddas@tezu.ernet.in)

(Received July 15, 2016; revised and accepted Mar. 11, 2017)

Abstract

Offensive operations have been promoted by the aggressors using computer as a tool or target, resulting, a cyber attack in *web-application*s of an organization or the infrastructure of entire nation. Depending upon the attacker's target, one can classify some of the mostly occurred cyber attacks into five broad categories. It reports some of the common methods adopted in conducting these attacks and their defending techniques. This paper mainly address the possibility of cyber attacks due to the execution of malicious or unintended nature of scripts. It formulates a verification method of web document and perform experiment in the client-side using its benign *script* structure. This method is capable of detecting any malicious script which inserts in the web-document during transportation from server to the client or due to the previously stored content in the client or server operation. Satisfactory results have been found with the own-generated and publicly available data-set.

Keywords: Benign Logical Structure; Classification; Cyber Attack; Malicious Script Insertion; Web-Application

1 Introduction

Today's literate population is becoming totally dependent on accessing web applications using browser and performing activities such as - email, banking, domestic appliances etc. The advancement of web technology helps software developer in developing user friendly applications so that user can work with those applications easily. A large number of organizational documents and non-web based applications being transferred into web based applications so that the efficiency and effectiveness of accessing organizational data over the Internet can be improved. Using every latest technology, in one way software developers are increasing simplicity in working with these applications, other way, these applications becoming targets of intruders or malicious users.

A major security issue arises specially during accessing such applications, as a large number of attacks are possible if the vulnerability in those applications are not properly addressed. Attackers, intrude logically into an application by misusing those vulnerabilities and they try to disrupt its normal functioning. Researchers, have been working largely to address web security threats and they could achieve partially to address the issues. Intruders also get cleverer with time and they formulate newer ways of attack which helps in bypassing the security mechanism of an application. To formulate a concrete solution, researchers need more work on every possibility that may lead a successful attack. In this paper first, it presents various categories of cyber attacks and present practices of their defense. Second, it formulates and effective verification of web document. The method detects malicious scripts contained in a response page or web document, which are responsible for defacing user access.

Finally, it reports test results based on experimental evaluation. Satisfactory results have been found for a series of evaluation using the own-generated and publicly available data-set.

2 Security Risks in Accessing Web Applications

In a typical scenario, accessing an application software using web-browser over the *Internet* is shown in Figure 1. The sequence of user request for a static resource (e.g., organization's web-site or web-page), stored in the web server and its response is shown in Lines 1, 2, 3, 4.

However, user may requests for the processed details e.g., students' *grade card* containing the information about grades secured on various subjects and the total grade point earned (both cumulative and semester).

It can be retrieved by accessing application page stored in the *application server* followed by the database processing with the parameter values - *username*, *password*, student's *roll number* and *term* (*say* Spring/Autumn, Year). The response page or web-document generated by the application may contain some client-sided *script*(s). In the client-side, the *script*s perform validation on the user input data and send it to the originating server. The appli-

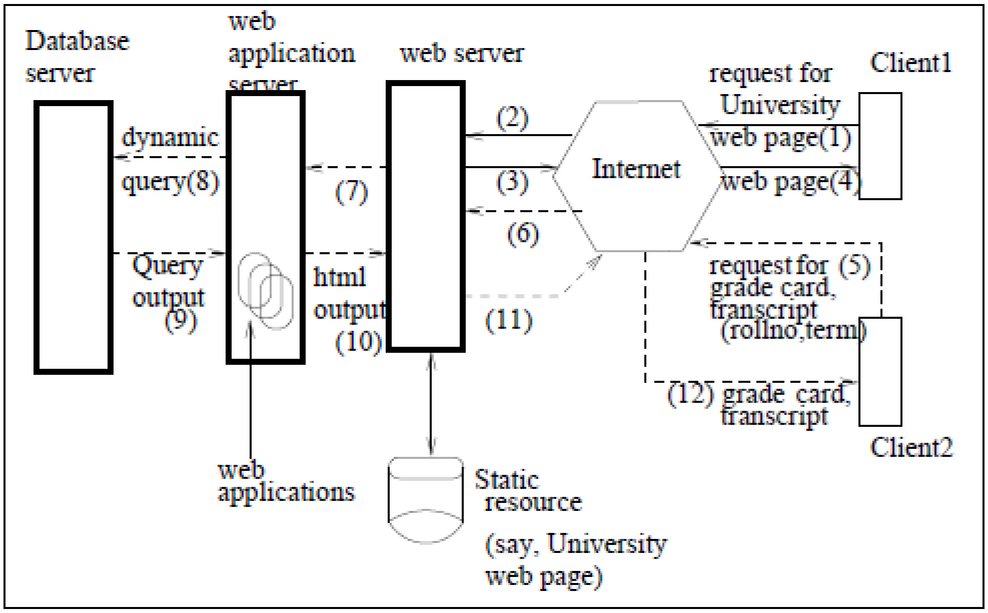


Figure 1: Typical Web application access

cation performs server-side processing using the parameter values, generates reports or web document (grade-card) and send it to the client over the Internet.

User requests for *grade-card* may come randomly to the application server over the *Internet*. For each request, the response page from the server generates one dynamic script. In the client, such scripts perform validation on user input - rollno, term_year and term_type (in Figure 1, dotted lines are shown - 5, 6, 7, 8, 9, 10, 11, 12). When such validation performs in the server, based on the generated scripts, it is called server side validation. The effectiveness of input validation can be increased by performing such at the both end. In Figure 2, it is shown how a legitimate access can be defaced due to the insertion of attack payload so that the legitimate access may re-direct to the attacker's site. Defacing user access mainly occurs due to the change in logical structure of a web-document by inserting the malicious script into it. During legitimate access the input parameter values or other session details such as - *user name*, *password* may be redirected to the attacker's site so that output of such application (i.e., grade-card) can also be redirected from the application server to the attacker's site without user's knowledge.

2.1 Contribution

We introduce a method of detecting *script vulnerabilities* causing cyber attacks based on verifying web document using its benign structure. We report various classes of cyber attacks and the identify the harmfulness of those. The deficiencies of existing detection techniques and the present day importance in addressing scripting vulnerabilities are highlighted. The effectiveness of the proposed method in real-life applications is also reported.

2.2 Cyber Attacks and Their Categories

Cyber attack may be defined as an offensive exercise employed by individuals or group that targets - information systems, infrastructures, computer networks, and/or personal computer devices. It generally, originate from an anonymous source and its main objective is to steal, alter, or destroy to a specified target. Based on their targets cyber attacks can be classified into five different categories described in Table 1.

In EPW, attackers mislead network users with false information by masquerading as a trustworthy entity. They mostly, misguide users through telephonic communication or pull to surf some decorated web sites loaded with false information. By surfing such or responding to their telephonic call, user may fall into the attacker's trap. In UKANF category, attacker tries to gather information on an application network by going through the organization's web pages or by executing some network commands that are commonly used for legitimate access. Attackers misuse vulnerability in a *web-application* under the MAV category and maliciously enter into the system so that they try to execute some unintended scripts or queries resulting undue knowledge of database, data-store, session details etc.

By running a successful MAV attack, attackers can manage to change the integrity of a data-store. Under MSVA category of attack, attacker tries to make the computer system or browser vulnerable by maliciously executing malware or virus or worms. By doing so, attackers try to redirect user access into a malicious *web-application* site without his/her knowledge. The DDoS category of attacks mainly targets network or computer resources. It generally executes by flooding with auto-generated re-

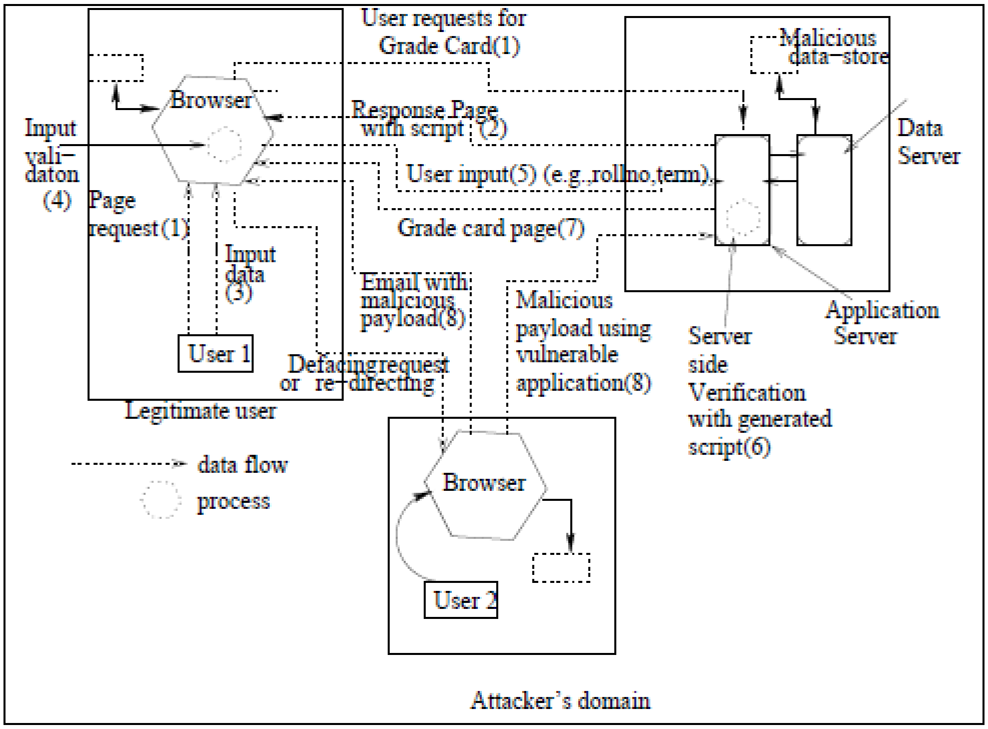

User requests for Grade Card(1)
Malicious data−store
Browser
Response Page with script (2)
Data Server
Input vali− daton (4)
User input(5) (e.g. rollno,term)
Page request(1)
Grade card page(7)
Input data (3)
Email with malicious payload(8)
Malicious payload using vulnerable application(8)
Application Server
Server side Verification with generated script(6)
User 1
Defacing request or re−directing
Legitimate user
data flow
process
Browser
User 2
Attacker's domain

Figure 2: Typical attack scenario

quests through botnet or some network configured specially for this purpose. To amplify the power of such attack, it generally implements in a distributed way with compromised hosts. Denial of service occurs to the legitimate users when congestion arise in the network bandwidth or the bandwidth of targeted web server's - CPU or disk or database.

2.3 Related Work

Under MAV category of cyber attack where, an attacker attempts to bypass authentication mechanism or steal session details without the knowledge of legitimate users. Insertion of unintended or malicious script in a web document during the execution of user access, is an effective method deploying by the attackers. Several solutions have been proposed by the researchers to address this menace so that the scripts execution is restricted in both client and server-side application. However, such restriction may effect application's functionality. Attack *script*s are prepared and injected into web document retrieves session details or deface user access or to perform other malicious operations. To overcome the scripting vulnerabilities, some of the defending coding practices are reported in [5, 18, 19]. Input validity checking for possible scripting vulnerabilities using predefined constraint, such as - black list, white list and syntax grammar, are popularly applied my most of the researchers [3, 17, 21, 23, 26].

The method of identifying vulnerabilities in a *web-application* and deploy defensive coding practices is proposed by some of the researchers [8, 12, 16]. Defensive coding approach is the primary requirement in preventing scripting vulnerabilities. However, dynamic *script* validation is equally important to prevent scripting attacks. To defend from typical scripting attacks, researchers proposed methods which avoid the process of escaping characters during execution of a web document [5, 18, 19].

In spite of implementing the defensive coding approach by the researchers it is found that the malicious *script*s execution comes to be the biggest threat in cyber world. Executing such *script* attackers may success in - defacing user access, stealing session details, changing the logical structure of web document, storing malicious strings in the machine etc. As a result, attackers can proceed for targeting computers or users in the social network.

The property of dynamic script generation in a *web-application* domain can be misused by the attackers, leading to scripting attacks. The current detection techniques [6, 9, 24] attempted to identify the vulnerable scripts in a web document that are dynamically generated. In a *web-application* domain, the method of bounding the script execution within the legitimate frame-work is proposed by some of the researchers [2, 10, 15]. However, such constraint may restricts application's functionality. Parse tree based verification technique to identify deviation between request and response is carried out by some of the researchers [1, 25]. However, due to the modified and enhanced crafting mechanism of *script* injection adopted by the attackers, it needs continuous effort for upgraded solution.

Table 1: Detection approaches and their pros and cons

Attack Class	Attack Method	Common Attacks	Defense
1. Exploit Psycho Weakness (EPW)	Exploring with pretext	Access, Phishing, Porno	Password restriction in login, filtering
2. Undue Knowledge of application & Network Finger-Print (UKANF)	Unintended actions - exploring networks, security scans and network audit	Reconnaissance, port scan, trust, MANNET passive, storage	Data encrypt, Rule based defense
3. Misuse Application's Vulnerability (MAV)	Insert/Execute malicious payload	Hacking, SQL injection, Cross-site scripting, Forgery, MANNET Active	Filtering input data, Firewall/rule-based protection, web-document verification
4. Makes System Vulnerable to Attacks (MSVA)	Tools: Crack-Security Sniffer, malware through stealth viruses, Trojan, Piggy-back	Spam, Hacking espionage, Cyber terrorism Cyber war, black hole, MANNET Active	email filter, Dynamic Firewall or rule-based protection
5. Distributed Denial of Service (DDoS)	Distributed flooding to targeted network or device using bot traffic	DDoS, Application layer DDoS	Deploy IDS, IPS at strategic point

Authentication mechanism is explored to identify vulnerabilities and formulated enhanced technique of authentication based on personalized policies, maintained by the application server [13, 27, 31]. To avoid unauthorized access, researchers have proposed authority-based delegation of encryption mechanism for multi-proxy signature scheme [7, 14, 20]. Proxy-based delegation mechanism for signature based verification is widely explored by the researchers to protect from unintended script execution [28, 29, 30].

3 Motivation

Insertion of malicious *script*s in a web document called *script* injection, may cause some notorious *web-application* attacks, such as - cross-site scripting(XSS), cross-site request forgery etc, To illustrate *script* injection, let us consider the typical *web application* access scenario as shown in Figure 1. The application is a simple on-line *student information system* (SIS) that allows to view academic details of students' admitted in various programme.

In a typical application report, student retrieves his/her grade-card containing information on various subjects in which he/she has registered, grade secured, grade point earned, semester, to credit completed etc. To view such details, user must log on to the proxy server (LINUX O.S.) configured in our University intranet (www.tezu.ernet.in). After successful login, it disconnects from such server and redirect the connection to the actual application server where SIS package (developed with PHP) exist. User must enter his/her valid login details (username and password) in the main page of this package. Login verification is done by a function module developed using the frame-work called *PHP Telnet*. The module retrieves password *script* file of the proxy server, containing account details and verify accordingly.

After successful login, the application generates a response page containing client-sided scripts and send the page to the client machine. The script verify user inputs - roll number or name or registration number and term (spring or autumn) so that by verifying user input with client-sided scripts, these are send to the server. Server-side application processes with these parameter values and generates response page or grade-card. During the process of user access if any unintended script containing some Java functions embedded in the web document, it will also execute. This unintended script may generate due to the previously store malicious content in the client or server side. It can also generate during communication with the previously stored malicious strings. It can be possible if user is not careful in clicking an application or user does not have awareness in accessing cyber-world. Due to the insertion of malicious payload or unintended client-sided script, user access to the application may be defaced because of the followings:

1) Change in logical structure of the web document;
2) Redirect the displayed page to the malicious site;
3) Session details stored in the cookies, send to the malicious site;
4) Malicious data-store.

The logical structure of accessing a web-document or document object model (DOM) defines how it is accessed and manipulated. DOM is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a web document. Let us consider a client-sided script file containing validity function to verify the input value of student’s roll number as given in Figure 3. In this function, it checks whether the length of input string for rollno is equal to 8 or not. If so, the string value is posted to the server in which the server page *abcd.phd* executes by taking this value as argument.

```
function rollnoFunction(){
var currentXhr=null; //Global variable is declared
$(”#roll_no”).keyup(function()){
var string = $(this).val();
//input value is stored in string variable
var word = string.toUpperCase(); //case conversion
if (word.length=1) {
currentXhr=$.ajax( {
type: ”POST”,
url: ”abcd.php”,
data:’keyword=’+word,
beforeSend: function(){
$(”#roll_no”).css(”background”,”#FFF url(Circle.gif)
no-repeat 380px”);
if(currentXhr != null && currentXhr.readyState!=4) {
currentXhr.abort();}},
success: function(data){
$(”#suggesstion-1”).show();
$(”#myModal”).modal(”show”);
$(”#suggesstion-1”).html(data);
$(”#roll_no”).css(”background”,”#FFF”); }) } }
else
$(”#suggesstion-1”).hide();
$(”#myModal”).modal(”hide”); }) }
```

Figure 3: Input string (Rollno) validation

Due to the insertion of malicious payload or unintended functional statement (say, Javascript function in a typical application) in the script file of web document, user access may executes in an unexpected manner. The displayed page may redirect to the malicious site and it may post some unintended script function injected into the web document. If the malicious payload containing function as given in Figure 4 say, *setCookie* (function to store user name in cookie variable) and *getCookie* (function to return the value of specified cookie).

There may occur some unintended action of sending information from client to the attacker’s site. This can be done by executing the script code containing *url* of attacker’s site concatenated with the string as - http://www.xul.fr/vulnera ble.html? cookie=”cookie_name”&cookie_valuse=name.

To prevent such kind of unintended execution, filtering of *script* from a web document is the common practices based on black list. Some of the defense method also restricts *script* in a web document. However, filtering mechanism or such restrictions may effect functionality of an application. In a *script* functional statements are mainly applied for client sided validation. Intended *script* functions available in a web document may be poisoned due to the unintended action. Thus, before executing script functions of a web document it must be verified based on legitimate structure.

```
function setCookie(cookie_name,cookie_val,exp_days)
{
var d = new Date();
d.setTime(d.getTime() + (exp_days*24*60*60*1000));
var expires = ”expires=”+d.toUTCString();
document.cookie = cookie_name + ”=” + cookie_val
+ ”; ” + expires;
}
function getCookie(cookie_name) {
var name = cookie_name + ”=”;
var ca = document.cookie.split(’;’);
for(var i = 0; i < ca.length; i++) {
var c = ca[i];
while (c.charAt(0) == ’ ’) {
c = c.substring(1);
}
if (c.indexOf(name) == 0) {
return c.substring(name.length, c.length);
}
}
return ””;
}
```

Figure 4: Unintended script to set and return cookie value

4 Proposed Approach

A response page or web document send by the *web-application* server against user request is associated with its *benign script structure* called *script proof*. The *script proof* is attached with the page so that before browsing a web document, its structure is verified. The detection approach is implemented in co-ordination with application server and client’s web browser. A program module retrieves structure of various *script*s available in a web document or response page and embed into it before sending to the client’s end. Before browsing a web document in the client machine, a plugged-in detection module retrieves - (1) its structure (excluding *script proof* part) called *candidate proof* and (2) the attached *script proof*. The module finally verify the two proofs so that if deviation between them is found, it is considered as an unsafe or vulnerable execution that may cause a scripting attack. The proposed idea is to dynamically construct the *script proof* whenever the application program constructs the response page against a web request. A *script proof* of a response page is considered to be self-evidently non-attacking so that any new insertion of *script* during transportation or due to the previously stored malicious content, a web document can be verified with its *script proof*.

4.1 Problem Formulations

Accessing a *web-application* is comprised of the following steps relevant to scripting attacks:

1) It generates dynamic *script*s based on user request;

2) The generated *script*s are embedded into the response page;

3) User inputs are verified based on the instructions on the *script*s;

4) While browsing a web document or response page, it does not verify the nature of *scripts/scripts function* - intended or malicious;

5) Both server and the client do not maintain the details of the scripts embedded into a response page against an user request;

6) A malicious script may inject into a response page due to the previously stored to the careless attempt by the user;

7) The logical structure of the response page or web document may be effected due to the insertion of malicious scripts stored in the client's machine;

8) Browsing of web document may be defaced due to the non-persistent strings other than as mentioned in Items 6) and 7);

4.2 Definitions

Web-application: A *web-application* is comprised of a set of *web-document*s written in web language such as - *html*, ASP, JSP, API, PHP etc. Each *web-document* can be expressed as an ordered set of - scripts, *html* strings and input-output data. A *web-document*(F_w) maps one or more out-going run-time expressions(Ri_{exp}).

A *web-document* can be expressed as -

$$\begin{aligned} W_d: \quad & (\{h_1, h_2, h_3, \cdots, h_m\} \cup \{s_1, s_2, s_3, \cdots, s_k\} \\ & \cup \{d_1, d_2, \cdots, d_n\}), \end{aligned}$$

where, $\{s_1, s_2, \cdots, s_k\} \leftarrow$ set of all scripts;

$\{h_1, h_2, h_3, \cdots, h_m\} \leftarrow$ set of all *html* string;

$\{d_1, d_2, \cdots, d_n\} \leftarrow$ input(1) or output(o) data (user input data or data generated by the web-application server);

The mapping of a server-sided *web-document*s of a *web-application*(F_w) into a response page (R_p) can be expressed as -

$$\begin{aligned} R_p &= \langle H \cup S \cup D \rangle \\ &= \langle \{h_1, h_2, \cdots, h_k\} \cup \{s_1, s_2, s_3, \cdots, s_m\} \\ &\quad \cup \{o_1, o_2, \cdots, o_n\} \rangle; \end{aligned}$$

Legitimate web-document: A web-document (W_d) is defined as *legitimate* or *normal* if its *candidate proof* match with the *script proof*. The function $f_b(S)$ retrieves the *script proof* (B_S) of *web-document* W_d containing scripts $S = \{s_1, s_2, s_3, \cdots, s_k\}$. $f_b(S) = B_S$, embed into the web document or response page (R_p) from the server so that the final web document passed from server to the client, can be expressed as: $R_p^{'} = \langle R_p, B_S \rangle$. The function f_c of the detection module retrieves *candidate proof* C_S from R_p containing the *script*s, $S^{'} = \{s_1^{'}, s_2^{'}, s_3^{'}, \cdots, s_k^{'}\}$, $f_b(S^{'}) = B_S$. Thus, a test web document is legitimate if $C_S = B_S$.

Malicious or attack web-document: A web-document (W_d) is defined as *malicious* or *attack* if its benign structure or *script proof* does not match with its *candidate proof*, i.e., $C_S \neq B_s(R_p)$.

The *benign structure* or *script proof* of a *web-document* is the sequence of its scripts structure. The structure of a *script* is the sequence of its *functional statement*s or *web language function*s, e.g., the benign structure of the function *getCookie(cookie_name)* is *document.cookie.split,return* as shown in Figure 4. In the application server, an auto-generated module generates the *script proof* of a response page or *web-document* before sending it to the client. In the client, before browsing a *web-document* using client's *web-browser* its structure is verified with its *script proof* using detection module attached to the web browser.

4.3 Detection Algorithm

The detection of nature of a *web-document* as *normal* or *malicious* is done using two different algorithms. First algorithm *web-document-structure*, return structure of a *web-document* by retrieving each script structure. The input to this algorithm is - *web-document* and list of web language functions. The output of this algorithm is the *script* structure of web document, containing one or more sub-strings separated with coma(,). Each sub-string is a web-language function. The second algorithm *exact matching*, retrieves the embedded sequence from the *web-document* called *script proof* and match it with *candidate proof* of the web document. The input to this algorithm is the response page or test *web-document*. The algorithm performs exact matching between the two strings, so that if it is found matched it gives output as *normal*, otherwise, declare as *attack*.

5 Experimental Results

To experiment the proposed detection approach, a three-tire platform is configured having database server at the back-end and web-application server in the middle, as shown in Figure 1. A small web-application is developed to maintain students' examination database and to

Table 2: Matching results of benign and candidate structure of script

Ex.	Benign Structure	Candidate Structure	Malicious String	attack/normal
1	Keyup, val, to Uppercase ajax, css, abort, show, modal, html, css	Keyup, val, to Uppercase ajax, css, abort, show modal, html css	nil	normal
2	Keyup, val, to Uppercase ajax, css, abort, show, modal, html, css	Keyup, val, to Uppercase, ajax, css, abort, show, modal, html css, cookie, split, substring	cookie, split, substring	attack

retrieve their academic information - result sheet, grade card and transcript. Accordingly, the application generates three response pages or web-document based on user request.

In the server side operation, each page containing two functional scripts, in which, the first two page containing scripts having validation code to validate user input data - roll number, term year, term type (Spring or Autumn). The third page containing script to validate user input value for the roll number or registration number. An application program executes to retrieve *script proof* of the generated web document and embedding into it before executing the send command. Each of the page containing one additional *script* for keeping *script proof*.

In the client-side operation, a plug-in module called scripts verification (SV), verifies the response page before browsing the actual page content. The module performs three operations - (1) it retrieves *script proof* of the page; (2) generates scripts structure (*candidate proof* of the page by sequentially retrieving each *script* other than the *script* containing *script proof*; (3) performs exact matching between *candidate proof* and *script proof* and gives output message based on the matching result.

In Figure 4, two malicious scripts containing unintended or *attack* Java function *document.cookie()*, *document.cookie.split(';')* and *return c.substring(name.length, c.length)*. These are mainly applied for printing the cookie details, having session information. This information can be redirected to the attacker's site so that without the knowledge of legitimate user, information may be retrieved by the attacker. The proposed detection module is tested with a series of web-document having *normal* and *attack* scripts written with Java functions. The proposed detection module also tested using some of the publicly available data-set containing malicious scripts [4, 11, 22]. Using these publicly available attack scripts, simulation of test module has been done with 20 successive operations against each response page or web-document. In each operation, one response page is tested by inserting one and more malicious scripts available in the data-set. Satisfactory results has been found in every operation. In Table 2, sample results of some of the client-sided experiment are shown. In this Table 2, the *script proof* retrieved from the web-document based on the sequence of functions available in the script. The first document containing one *script*, accordingly, all the functions starts with dot (.), other than the statement define file with the extension dot(.), i.e., while retrieving from first document, it is not included the statement *url (blueCircle.gif)*. The sequence of *function*s in the respective proof are kept separated with delimiter coma (,). While testing with public data-set, we did not find any false positive.

In Table 2, the sample results are shown based on our experiment using a typical application with limited numbers of user access. In a largely accessible web application, where, in generates different run-time scripts for every response page, the proposed method of generating benign *script* structure and embedding as *script proof* into it, may effects the performance of web application access. To improve the efficiency, an indexing technique is applied to a sequence of probable *script proof*s generated during training phase. The index of a script structure is prepared by considering first 3 characters of each Java function available in it. At run-time, the server-side module retrieves the appropriate *script proof* of a response page using its index. To test the indexing technique, separate applications are developed so that for every legitimate user request it generates different run-time scripts. The proposed indexing-based verification process is tested with 100 numbers of distinct run-time scripts. Using indexing technique, performance of proposed method has been improved while testing at run-time. Depending upon the accessibility and dimension of web applications, the performance of proposed method can be upgraded using multiple indexing technique. A sample shot of indexed *script proof*s are shown in Table 3. A comparative report based on results found in series of experiments is reported in Table 4.

6 Conclusion

Scripting attacks due to the unintended functional statement in the web document is considered to be one of dangerous threats to the web enabled applications. To avoid such attack, both application developer and user must aware with this attack and accordingly, they must be careful in developing software and accessing the same

Table 3: Index of script structures

Sl	Index	Script Structure
1	KeyvaltoUajacssa boshomodhtmcss	Keyup, val, toUppercase, ajax, css, abort, show, modal, html, css
2	Keyvalajacssa boshomodhtmcss	Keyup, val, ajax, css, abort, show, modal, html, css
3	KeyvaltoUajacs sabomodhtmcss	Keyup, val, toUppercase, ajax, css, abort, modal, html, css
4	Keyvalajacssabo shomadhtlcss coosplsub	Keyup, val, toUppercase, ajax, css, abort, show, modal, html, css, cookie, split, substring

Algorithm 1 Function to retrieve structure of a script

```
Functionweb-document-structureInput:$list_web_functions(L)$,$web - document(W_d)$ Output:$structureW_s[]$
Begin
$W_s[] \leftarrow 0$;
while not all $scripts$ in $W_d$ done do
  Read script S;
  while not all $substrings$ in $S$ done do
    Retrieve a sub-string $W_s$ from $W_d$;
    while not functions in $L$ from 1 to $n$ done do
      Read a function in $L \leftarrow f_w$;
      if $W_s = f_w$ then
        $W_s[] \leftarrow concat(W_s[], W_s)$;
      end if
    end while
  end while
end while
Return $(W_s[])$;
EndFunction
End
```

Algorithm 2 Script structure matching

```
Function exact-matchingInput:Test web-document $R_p$ Output:page-status
begin
Read $R_p$;
Retrieve benign structure $B_s$ from $R_p$;
Cut portion of $B_s$ from $R_p \leftarrow Ri_p^{'}$
$C_s$ = call $web - document - structure(R_p)$;
if $C_s = B_s$ then
  page-status='Normal web-document';
else
  page-status='Malicious or Attack web-document';
end if
Return (page-status);
EndFunction
end
```

respectively. In the server-side operation, the previously stored malicious content that need to be addressed. In a typical method, the *script proof* must be generated before retrieving the stored data that effects in constructing the *script*s of a web document. This is due to the fact that if the previously stored data is a malicious content than the *script proof* also will be effected. However, this may not be possible always, as, there can be applications where scripts in web document generates dynamically with the stored data. To avoid such situation where the script generates using the stored content the filtering of stored content or retrieved data must be required based on the initialized constraint. However, in the client-side application this method has no limitations and can be applied directly to avoid *script* attacks. However, if the dynamically generated *script*s in the response page are due to the multiple *web-application* servers before coming to the client's machine, then the proposed method need to be explored with signature generation and verification in each server.

Acknowledgments

This work is funded by Ministry of IT, Govt of India under the scheme of ISEA, Phase II.

References

[1] E. Athanasopoulos, A. Krithinakis, and E. P. Markatos, "Hunting cross-site scripting attacks in the network," in *Proceedings of the 4th Workshop on Web 2.0 Security Privacy (W2SP'10)*, pp. 1–8, 2010.

[2] T. S. Barhoom and S. N. Kohail, "New server-side solution for detecting cross site scripting attack," *International Journal of Computer Information Systems*, vol. 3, no. 2, 2011.

[3] P. Bisht and V. N. Venkatakrishnan, "XSS guard: Precise dynamic prevention of cross-site scripting attacks," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 23–43, 2008.

[4] Blwood, *Multiple XSS Vulnerabilities Intikiwiki 1.9x, Mailing List Bugtrap*, May 2006. (http://www.securityfocus.com/archive/1/435127/30/120/threaded)

[5] CWE-79, *CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')*, June 2010. (http://cwe.mitre.org/data/definitions/79.html)

[6] J. H. Hayes and A. J. Offutt, "Input validation analysis and testing," *Empirical Software Engineerings*, vol. 11, no. 4, pp. 493–522, 2006.

[7] M. S. Hwang, S. F. Tzeng, and S. F. Chiou, "A non-repudiable multi-proxy multi-signature scheme," *In-*

Table 4: Comparative results based on simulation

Method	Constraint for application functionality	Implementation Pros and Cons	Average DR using public/own data
Third party black list with dynamic update (IMPERVA 2012)	Exists	(1) Research oriented taint & white lists (3) Depend on third party lists	90%
Bounding script execution [2, 10, 15]	Exists	(1) Reliable for script execution (2) Unable to handle typical application due to restrictions	80-90%
Initialize taint or trust codes in client or server [3, 17, 21, 23, 26]	Exists	(1)Reliable for script execution (2) Unable to handle typical application due to initialization	89%
Proposed Method	Not exists	(1)Independ-ent handling of server & client operations (2) Effective verification of malicious content with script proof	98%

novative Computing, Information and Control Express Letters, vol. 3, no. 3, pp. 259–264, 2009.

[8] N. Jovanovic, C. Kruegel, and E. Kirda, "Pixy: A static analysis tool for detecting web application vulnerabilities," in *IEEE Symposium on Security and Privacy*, pp. 263–268, 2006.

[9] A. Kiezun, P. J. Guo, K. Jayaraman, and M. D. Ernst, "Automatic creation of SQL injection and cross-site scripting attacks," in *Proceedings of the 31st International Conference on Software Engineering*, pp. 1–11, 2009.

[10] E. Kirdaa, N. Jovanovicb, C. Kruegelc, and G. Vignac, "Client-side cross-site scripting protection.," *Computer and Security*, vol. 28, no. 7, pp. 592–604, 2009.

[11] J. Kratzer, "Jspwiki multiple vulnerabilities. posting to the bugtrap mailinglist," Sept. 2007. (http://seclists.org/bugtrap/2007/Sep/0324.html)

[12] H. T. Le and P. K. K. Loh, "Identification of performance issues in contemporary black-box web application scanners in SQLI," in *Latest Advances in Information Science and Applications*, pp. 211–216, 2012.

[13] C. C. Lee, C. H. Liu, and M. S. Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[14] C. Lin, K. Lv Y. Li, and C. C. Chang, "Ciphertext-auditable identity-based encryption," *International Journal of Network Security*, vol. 17, no. 1, pp. 23–28, 2015.

[15] M. Madou, E. Lee, J. West, and B. Chess, "Watch what you write: Preventing cross-site scripting by observing program output," in *Application Security Conference*, pp. 1–14, 2008.

[16] M. Martin, B. Livshits, and M. S. Lam, "Finding application errors and security flaws using PQL: A program query language," in *Proceedings of the 20th Annual ACM SIGPLAN Conference*, pp. 1–19, 2005.

[17] K. K. Mookhey and N. Burghate, *Detection of SQL Injection and Cross-site Scripting Attack*, Mar. 17, 2004. (http://www.symantec.com/connect/articles/detection-sql-injection-and-cross-site-scripting-attacks)

[18] A. Mueller, *Cross Site Scripting (XSS)*, May 2009. (http://elegantcode.com/2009/05/28/cross-site-scripting-xss/)

[19] OWASP, *XSS (Cross Site Scripting) Prevention Cheat Sheet*, Jan. 2010. (http://www.owasp.org/index.php/XSS-Prevention-Cheat-Sheet)

[20] C. Pan, S. Li, Q. Zhu, C. Wang, and M. Zhang, "Notes on proxy signcryption and multi-proxy signature schemes," *International Journal of Network Security*, vol. 17, no. 1, pp. 29–33, 2015.

[21] R. Pelizzi and R. Sekar, *Protection, Usability and Improvements In Reflected XSS Filters*, 2012. (http://www.seclab.cs.sunysb.edu/seclab1/pubs/xss.pdf)

[22] A. Pigrelax, *XSS in Nested Tag in PHPBB 2.0.16. Mailing List Bugtrap*, July 2005. (http://www.securityfocus.com/archive/1/404300)

[23] R. Sekar, *An Efficient Black-box Technique for Defending Web-application Attacks*, Defense Advanced Research Project Agency, the Naval Surface Weapons Center, 2009.

[24] H. Shahriar and M. Zulkernine, "Mutec: Mutation-based testing of cross site scripting," in *Proceedings of the 5th International Workshop on Software Engineering for Secure Systems*, pp. 47-53, IEEE, 2009.

[25] J. Shanmugam and M. Ponnavaikko, "A solution to block cross site scripting vulnerabilities based on service oriented architecture," in *EEE/ACIS International Conference on Computer and Information Science (ICIS'07)*, 2007.

[26] Z. Su and G. Wassermann, "Static detection of cross-site scripting vulnerabilities," in *In 30th ACM/IEEE 30th International Conference on Software Engineering*, 2008.

[27] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: Current status and key issues," *International Journal of Network Security*, vol. 3, no. 2, pp. 101-115, 2006.

[28] S. F. Tzeng, M. S. Hwang, and C. Y. Yang, "An improvement of nonrepudiable threshold proxy signature scheme with known signers," *Computers and Security*, vol. 23, no. 2, pp. 174–178, 2004.

[29] S. F. Tzeng, C. C. Lee, and M. S. Hwang, "A batch verication for multiple proxy signature," *Parallel Processing Letters*, vol. 21, no. 1, 2011.

[30] S. F. Tzeng, C. Y. Yang, and M. S. Hwang, "A nonrepudiable threshold multi-proxy multi-signature scheme with shared verication," *Future Generation Computer Systems*, vol. 20, no. 5, pp. 887–893, 2004.

[31] H. C. Wu, M. S. Hwang, and C. H. Liu, "A secure strong-password authentication protocol," *Fundamenta Informaticae*, vol. 68, no. 4, pp. 399–406, 2005.

Biography

Debasish Das, received his Ph.D. in Computer Science and Engineering from Tezpur University in 2015 in the field of Network Security. He has been involved in application systems design & development and teaching for last 20 years. Presently, he is working as Systems Analyst in the Department of Computer Science and Engineering at Tezpur University. His research area includes Network and Information Security, Machine Learning and Financial Computing.

Dhruba Kr Bhattacharyya received his Ph.D. in Computer Science from Tezpur University in 1999 in the field of Cryptography and Error Control Coding. He has been in teaching for last 24 years and presently, he is a Professor at HAG in the Computer Science and Engineering Department at Tezpur University. His research areas include Machine Learning, Network Security and Bioinformatics. Prof. Bhattacharyya has published more than 240 research articles in the leading International Journals and peer-reviewed conference proceedings. Dr. Bhattacharyya also has written/edited 12 books. He is a fellow of IETE. Dr Bhattacharyya is on the editorial boards of several international journals and also on the programme committees/advisory bodies of several international conferences/workshops.

New Kind of Delegation-based Anonymous Authentication Scheme for Wireless Roaming Networks

Chun-Lin Jiang[1], Shi-Lan Wu[1], Ke Gu[2]

(Corresponding author: Chun-Lin Jiang)

School of Mathematics and Computer Sciences, Xinyu University[1]
Xinyu 338004, Jiangxi, China
(Email: 38029747@qq.com)
School of Computer and Communication Engineering, Changsha University of Science and Technology[2]
Changsha 410004, China

(Received Mar. 15, 2017; Revised and Accepted June 25 & July 13, 2017)

Abstract

In order to reduce message flows of traditional anonymous authentication schemes, a new kind of delegation-based scheme is proposed for wireless roaming networks. By making use of a proxy signature, the new scheme requires only a user and a visited server to participate in the authentication process, without the real-time participation of user's home server. Therefore, the new scheme needs less message flows than traditional schemes. In an instantiation of the new scheme, elliptic-curve cryptography (ECC) is used to keep efficiency, and the mobile station needs only 3.25 elliptic curve scalar multiplication (ECSM) operations, which are 5.5ECSM and 3Pairing less than the scheme based on group signature. The comparison shows that, though the unlinkability of our scheme is weaker, the computation load is much lower. So our scheme is efficient and practical.

Keywords: Anonymous Authentication; Delegation; Diffie-Hellman Key Exchange; Proxy Signature; Wireless Roaming Networks

1 Introduction

In wireless roaming networks, when a mobile station (MS) authenticates itself to a visited location register (VLR), the identity (ID) of MS is often valuable and must be protected. Because MS registers to its home location register (HLR), VLR often needs to communicate with HLR to authenticate MS, and MS also needs HLR to authenticate VLR.

The authentication process of most existing anonymous authentication protocols involves three parties including MS, VLR and HLR. According to the needed computational operations in MS, these traditional protocols are often divided into three types: (1) non-encryption based type that needs no cryptographic operations in MS [3, 22]; (2) secret-key based type that needs symmetric encryption operations in MS [9, 26]; (3) public-key based type that needs asymmetric encryption operations in MS [1, 6, 7, 10, 15, 25]. The first type often uses one-way hash functions and exclusive-OR operations to reduce the computation cost in MS, but it requires too many message flows (eight flows in [3]). The second type is a common type, but it cannot solve the non-repudiation and key management problems. The third type is a hotspot recently because it can provide non-repudiation and key management service, but it is computationally expensive even though hardware prices have fallen a lot. Besides, all three types need the real-time participation of HLR and require at least four message flows. It is well known that the bandwidth of wireless networks is limited, and VLR is often far from HLR, so the involvement of HLR often makes the communication time too long to bear.

Therefore, it is necessary to design authentication protocols involving only MS and VLR. Protocols based on group signatures [17, 24] or ring signatures [23] (actually a ring signature is a simplified group signature) can meet the requirement. In this kind of scheme, HLR is considered as the group manager of a group signature system and MS as a member of the group; when MS roams to VLR, MS signs messages on behalf of the group without showing its ID; by verifying the group signature, VLR is sure that MS is one valid user of HLR. Though this kind of scheme often needs only three message flows, it is still not practical for realistic applications because it is complex for MS to generate a group signature.

This paper proposes a new kind of delegation-based scheme which not only meets the requirement but also has good performance. The rest of this paper is organized as follows. Section 2 reviews some existing work on proxy signature. Section 3 introduces the system model of the new scheme. Section 4 gives two examples to instantiate the scheme. Finally, we analyze and conclude it in Sections 5 and 6.

2 Related Work

Mambo [16] gives a definition of proxy signature as follow.

Definition 1. *A proxy signature is a signature that is generated by a proxy signer on behalf of the original signer. It is often used in the following scenario: a manager delegates his/her signature authority to his/her trustworthy assistant in advance; when he/she is too far away to sign a document, the assistant has the power to sign it on behalf of the manager. It includes three types of delegation: full delegation; partial delegation; delegation by warrant.*

Proxy signatures were used to construct traditional anonymous authentication protocols involving three parties [2, 4, 5, 8, 11, 12, 13, 14, 19, 20, 21]. In 2005, Lee and Yeh [12] proposed an anonymous authentication protocol based on partial delegation for wireless communication system. Their protocol adopted the public-key system to achieve the security requirements and employed off-line authentication process to save authentication time. But Lee and Chang [11] showed that it could not achieve non-repudiation in off-line authentication process. They presented an improved protocol which not only avoided the weakness but also reduced the computation cost. In 2008, Tang *et al.* [19] proposed an efficient anonymous authentication protocol based on delegation by warrant for wireless networks. The protocol uses elliptic-curve cryptography (ECC) to ensure safety and efficiency. But in 2014, Kumar *et al.* [8] demonstrated that Tang-Wu's scheme did not achieve the user unlinkability. They then proposed a robust authentication model utilizing the biometric to get unlinkability.

The authentication process of the above protocols can be summarized as follows. First HLR authorizes MS the power to sign; when MS roams to VLR, MS computes a valid proxy signature without showing its real ID; then VLR verifies the legality of MS based on the public key of HLR; finally HLR authenticates VLR and generates a session key for MS and VLR. Just as mentioned above, the real-time participation of HLR results in many message flows: five flows are needed in [5], six flows are needed in [12, 11] and four flows are needed in [19, 8]. Actually HLR is removable with two reasons. Firstly, MS can also authenticate VLR by verifying the signature of VLR. Secondly, according to [24], the session key should be only known to MS and VLR, and should be derived from contributions of both of them; in particular, HLR should not generate it for them. So it is feasible and necessary to change these protocols to the scheme involving only MS and VLR.

3 System Model

Our new scheme is described as follows. first HLR delegates his signature authority to MS in advance; when MS roams to VLR, VLR computes an ordinary signature and sends it to MS; then MS authenticates VLR by verifying the signature; finally MS computes a valid proxy signature without showing its real ID and VLR authenticates MS based on the public key of HLR. During the authentication, a session key is derived from MS and VLR.

Our scheme is composed of three parts: Initialization, delegation, and authentication.

1) Initialization: Let ID_M, ID_V and ID_H be ID of MS, VLR and HLR respectively; $Sig()$ and $Verify()$ be the signing and verifying algorithms of an ordinary signature scheme such as digital signature algorithm (DSA); $PSig()$ and $PVerify()$ be signing and verifying algorithms of a proxy signature scheme respectively. VLR has a private/public key pair (x_V, y_V).

2) Delegation: HLR generates a pseudonym $alias$ and a proxy signing key x_p for MS. The proxy verifying key y_p, which is often HLR's public key, is put in public by HLR.

3) Authentication: When MS roams to VLR, the authentication process between MS and VLR is in Figure 1.

It is illustrated as follows.

1) MS sends $alias$ to VLR.

2) VLR generates an ordinary signature σ_v on message m_v, and sends (m_v, σ_v, ID_v) to MS.

3) MS verifies σ_v with VLR's public key y_v. If the signature is valid, MS computes a proxy signature σ_M on message m_M, and then sends (m_M, σ_M, ID_H) to VLR. Otherwise, it rejects the connection.

4) VLR verifies σ_M with y_p. If the signature is valid, it accepts the connection. Otherwise, it rejects the connection. During the authentication, a session key is derived from m_M and m_v.

4 Two Examples

4.1 An Example Based on Partial Delegation

Lee and Chang's protocol [11] includes on-line and off-line authentication processes. It uses a backward hash chain to ensure the security, but it still has some weaknesses.

4.1.1 Review of Lee and Chang's Protocol

Figure 2 is the protocol of Lee and Chang. The protocol is illustrated as follows.

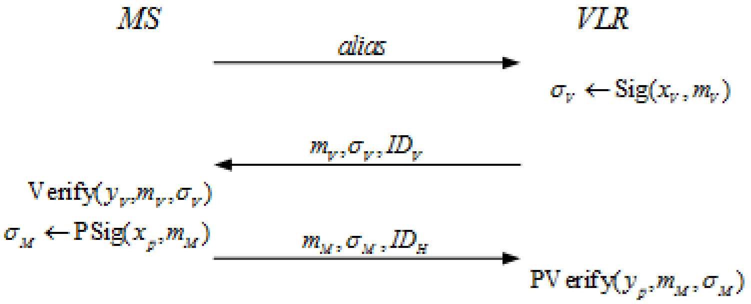

MS
VLR
alias
$\sigma_v \leftarrow \mathrm{Sig}(x_v, m_v)$
m_v, σ_v, ID_v
$\mathrm{Verify}(y_v, m_v, \sigma_v)$
$\sigma_M \leftarrow \mathrm{PSig}(x_p, m_M)$
m_M, σ_M, ID_H
$\mathrm{PVerify}(y_p, m_M, \sigma_M)$

Figure 1: The authentication process of our model

1) Initialization: Let Z_p^* be a group of large prime order p, g be a generator of it, and q be a prime factor of $p-1$; K_{HV} be a shared key between HLR and VLR; (x, v) be a private/public key pair of HLR, with x a random number and $v = g^x \bmod p$; $[M]_K$ be the encryption of M using a symmetric key K; $h()$ be a one-way hash function; $||$ be a concatenation operator.

2) Delegation: First HLR generates a random number k and computes $\sigma = x + kK \bmod q$ as MS's proxy signing key and $K = g^k \bmod p$ as MS's pseudonym. Then HLR stores (σ, K) in its database and gives them to MS simultaneously.

3) On-Line Authentication:

   a. MS selects a random number n, pre-computes $h^{(i)}(n_1), h^{(2)}(n_1), \cdots, h^{(n+1)}(n_1)$ with $h^{(i)}(n_1) = h(n_1)$ and $h^{(i+1)}(n_1) = h(h^{(i)}(n_1))$ for $i = 1, 2, \cdots, n$. It then sends K to VLR.

   b. VLR selects a random number n_2 and sends (n_2, ID_v) to MS.

   c. MS selects a random number t, sets $N_1 = h^{(n+1)}(n_1)$, and then computes $r = g^t \bmod p$ and $s = \sigma h(N_1||n_2||ID_v) + tr \bmod q$ as the proxy signature. It then sends $(r, s, K, N_1, ID_H, ID_v)$ to VLR.

   d. VLR verifies the signature by checking $g^s = (vK^K)^{h(N_1||n_2||ID_v)} r^r \bmod p$. If the equation holds, VLR sends $([N_1||n_2||K]_{K_{HV}}, ID_H, ID_v)$ to HLR. Otherwise, VLR rejects the connection.

   e. HLR decrypts $[N_1||n_2||K]_{K_{HV}}$ and gets K. It then gets σ from its database and selects a random number n_3 to compute a session key $C_1 = h(N_1||n_2||n_3||\sigma)$ for VLR and MS. Finally HLR sets $l = N_1$ and sends $([[N_1, n_3, ID_v]_\sigma||n_2||l||C_1]_{K_{HV}}, ID_H, ID_v)$ to VLR.

   f. VLR gets $[N_1, n_3, ID_v]_\sigma||n_2||l||C_1$, checks (n_2, l), and accepts C_1 as the session key. Then VLR sends $([N_1, n_3, ID_v]_\sigma, ID_v)$ to MS.

   g. MS decrypts $[N_1, n_3, ID_v]_\sigma$, checks N_1 and computes the session key C_1.

4) i^{th} Off-Line Authentication:

   a. MS computes $[h^{(n-i+1)}(n_1)]_{C_1}$ and sends it to VLR for $i = 1, 2, \cdots, n$.

   b. VLR checks $h(h^{(n-i+1)}(n_1)) = l$, sets $l = h^{(n-i+1)}(n_1)$ and computes the session key $C_{i+1} = h(l, C_i)$. It then updates $i = i + 1$ and checks $i \leq n$.

4.1.2 Analysis of the Protocol

The protocol is not efficient because it needs six message flows in on-line authentication process. It is also not secure because HLR knows the session key between VLR and MS.

4.1.3 Improved Protocol

Figure 3 is the improved protocol which is based on our model and is illustrated as follows.

1) Initialization: The same as original protocol. Besides, VLR has a key pair (x_v, y_v) of DSA.

2) Delegation: The same as original protocol.

3) On-Line Authentication:

   a. MS selects a random number n_1, pre-computes $h^{(i)}(n_1), h^{(2)}(n_1), \cdots, h^{(n+1)}(n_1)$ with $h^{(i)}(n_1) = h(n_1)$ and $h^{(i+1)}(n_1) = h(h^{(i)}(n_1))$ for $i = 1, 2, \cdots, n$. It then sends K to VLR.

   b. VLR selects a random number t_v and computes $n_2 = g^{t_v} \bmod p$. Then VLR computes a DSA signature σ_v on $n_2||ID_v$ and sends (n_2, σ_v, ID_v) to MS.

   c. MS verifies σ_v, selects a random number t and sets $N_1 = h^{(n+1)}(n_1)$. It then computes $r = g^t \bmod p$ and $s = \sigma h(N_1||n_2||ID_v) + tr \bmod q$ as the proxy signature. Finally MS sends $(r, s, K, N_1, ID_H, ID_v)$ to VLR and computes a session key $C_1 = n_2^t$.

   d. VLR verifies the signature by checking $g^s = (vK^K)^{h(N_1||n_2||ID_v)} r^r \bmod p$, if the equation holds, then VLR computes $C_1 = r^{t_v}$ and $l = N_1$. Otherwise, VLR rejects the connection.

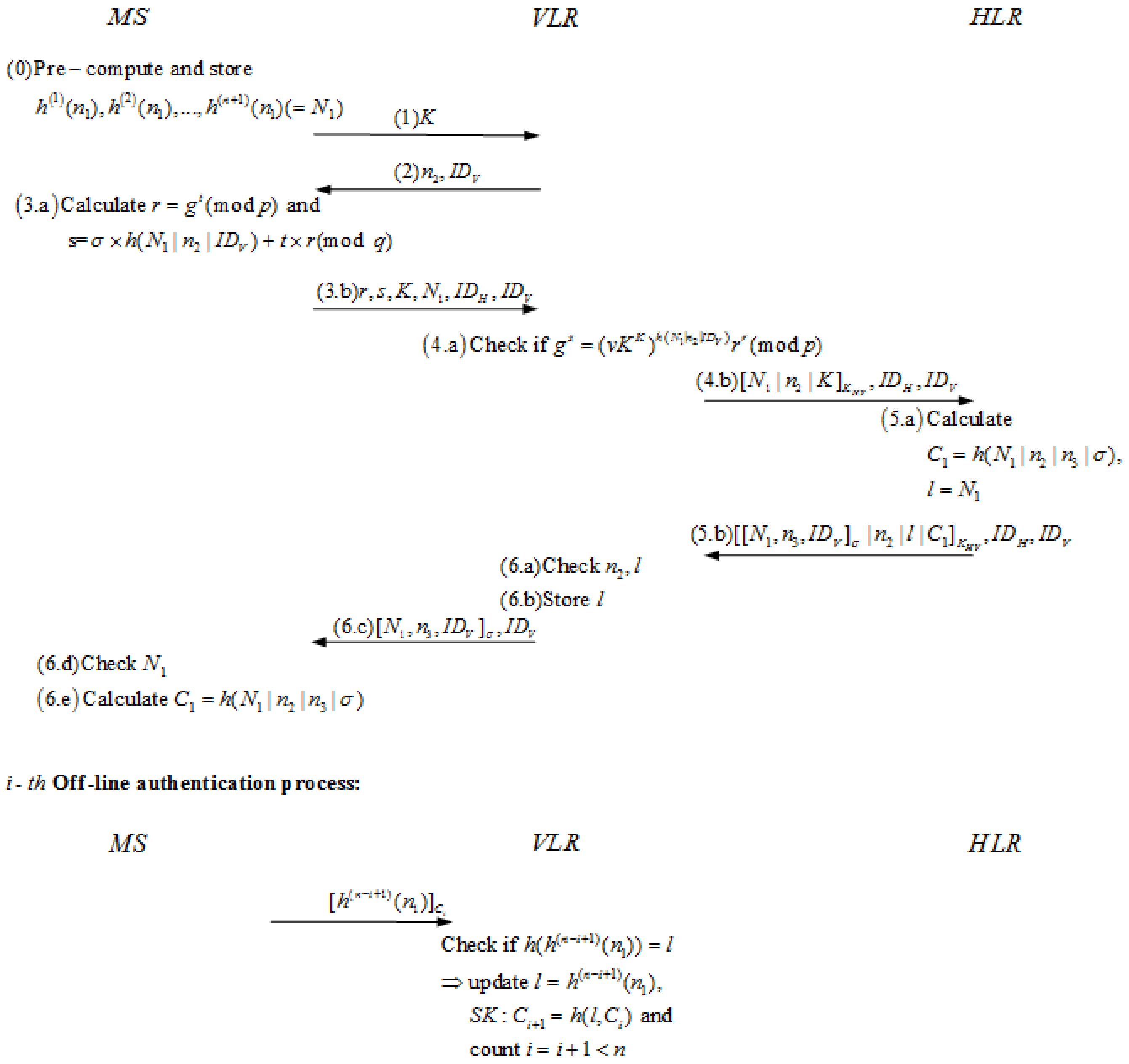

MS
VLR
HLR
(0)Pre – compute and store
$h^{(1)}(n_1), h^{(2)}(n_1), \ldots, h^{(n+1)}(n_1)(= N_1)$
(1)K
(2)n_2, ID_V
(3.a)Calculate $r = g^t (\bmod p)$ and
$s = \sigma \times h(N_1 | n_2 | ID_V) + t \times r (\bmod\ q)$
(3.b)r, s, K, N_1, ID_H, ID_V
(4.a)Check if $g^s = (vK^K)^{h(N_1|n_2|ID_V)} r^r (\bmod p)$
(4.b)$[N_1 | n_2 | K]_{K_{HV}}, ID_H, ID_V$
(5.a)Calculate
$C_1 = h(N_1 | n_2 | n_3 | \sigma)$,
$l = N_1$
(5.b)$[[N_1, n_3, ID_V]_\sigma | n_2 | l | C_1]_{K_{HV}}, ID_H, ID_V$
(6.a)Check n_2, l
(6.b)Store l
(6.c)$[N_1, n_3, ID_V]_\sigma, ID_V$
(6.d)Check N_1
(6.e)Calculate $C_1 = h(N_1 | n_2 | n_3 | \sigma)$
i - th Off-line authentication process:
MS
VLR
HLR
$[h^{(n-i+1)}(n_1)]_{C_i}$
Check if $h(h^{(n-i+1)}(n_1)) = l$
$\Rightarrow$ update $l = h^{(n-i+1)}(n_1)$,
$SK: C_{i+1} = h(l, C_i)$ and
count $i = i + 1 < n$

Figure 2: The authentication process of Lee and Chang

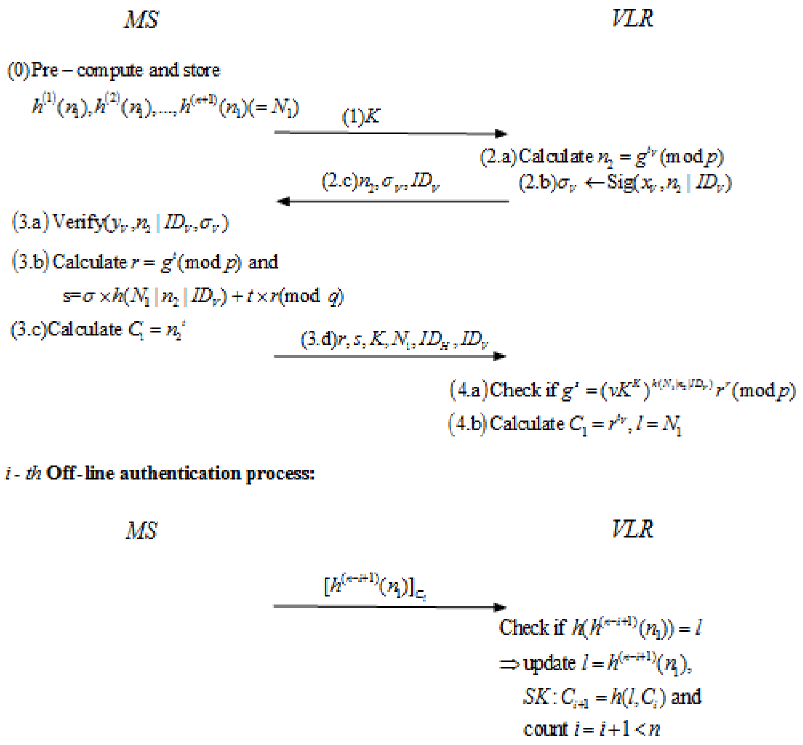

MS
VLR
(0)Pre – compute and store
$h^{(1)}(n_1), h^{(2)}(n_1), \ldots, h^{(n+1)}(n_1)(=N_1)$
(1)K
(2.a)Calculate $n_2 = g^{t_V} (\mathrm{mod}\, p)$
(2.b)$\sigma_V \leftarrow \mathrm{Sig}(x_V, n_2 \mid ID_V)$
(2.c)n_2, σ_V, ID_V
(3.a) Verify$(y_V, n_2 \mid ID_V, \sigma_V)$
(3.b) Calculate $r = g^t (\mathrm{mod}\, p)$ and
$s = \sigma \times h(N_1 \mid n_2 \mid ID_V) + t \times r (\mathrm{mod}\ q)$
(3.c)Calculate $C_1 = n_2^{\ t}$
(3.d)r, s, K, N_1, ID_H, ID_V
(4.a)Check if $g^s = (vK^K)^{h(N_1|n_2|ID_V)} r^r (\mathrm{mod}\, p)$
(4.b)Calculate $C_1 = r^{t_V}, l = N_1$
i - th Off-line authentication process:
MS
VLR
$[h^{(n-i+1)}(n_1)]_{C_i}$
Check if $h(h^{(n-i+1)}(n_1)) = l$
$\Rightarrow$ update $l = h^{(n-i+1)}(n_1)$,
$SK: C_{i+1} = h(l, C_i)$ and
count $i = i + 1 < n$

Figure 3: The improved authentication protocol

4) i^{th} Off-Line Authentication: The same as original protocol.

4.1.4 Performance Comparison

Table 1 is the performance comparison between original protocol and our improved protocol. Though MS needs one more calculation of public-key computation in our protocol, the message flows are greatly reduced from six to three in on-line authentication process. Besides, VLR and MS generate the session key based on Diffie-Hellman key exchange, which is secure under the decisional Diffie-Hellman (DDH) assumption. So our scheme is more efficient and secure.

4.2 An Example Based on Delegation by Warrant

In 2008, Tang *et al.* [18] proposed a proxy signature based on delegation by warrant by using an ECC system. We now combine this proxy signature with the elliptic curve digital signature algorithm (ECDSA) and the elliptic curve Diffie-Hellman (ECDH) exchange to design a really practical anonymous authentication protocol.

4.2.1 Description

Figure 4 is the protocol which is described as follows.

1) Initialization: Let F be a Galois field with an elliptic curve E in it, and T be a point of E; $(+)$ be a point addition operator in E; m_w be a warrant from which ID_M is not possible to be derived; Γ be public information used by VLR to verify MS; $\Pi()$ be a point representation function from E to Z_p; (x, Y) be a private/public key pair of HLR, with $x \in Z_p$ and $Y = xT$; (x_v, y_v) be an ECDSA private/public key pair of VLR; $h()$ be a secure hash function.

2) Delegation: HLR generates a pseudonym $IDMA = h(ID_M)$ for MS, selects a random number k and computes $\Gamma = (h(IDMA||m_w)T)(+)(kT)$ and $\sigma = -xh(\Pi(\Gamma)) - k$. Then HLR puts $(\Gamma, IDMA, m_w)$ in public, but delivers (σ, m_w) to MS secretly and securely. MS accepts the proxy signing key σ if $h(IDMA||m_w)T = (\sigma T)(+)(h(\Pi(\Gamma))Y)(+)\Gamma$.

3) Authentication:

   a. MS sends $IDMA$ to VLR.

   b. VLR selects a random number k_v, computes a ECDSA signature σ_v on $k_vT||ID_v$ and sends (k_vT, σ_v, ID_v) to MS.

   c. MS verifies σ_v, selects random numbers k and N, then computes $R = kT$ and $s = \sigma - kh(\Pi(R)||N)$ as the proxy signature. Finally MS sends (m_w, R, s, N, ID_H) to VLR and computes session key $C_1 = k(k_vT)$.

   d. VLR checks if (sT) $(+)$ Γ $(+)$ $(h(\Pi(\Gamma))Y)$ $(+)$ $(h(\Pi(R)||N)R) = h(IDMA||m_w)T$. If the equation holds, it then computes $C_1 = k_vR$ as

Table 1: Performance comparison between two protocols

Schemes	On/Off Line	Number of parties	Number of rounds	Number of secret-key computation in MS	Number of public-key computation in MS
Lee and Chang's protocol	On-line	3	6	1	1
	Off-line	2	1	n	0
Our improved protocol	On-line	2	3	1	2
	Off-line	2	1	n	0

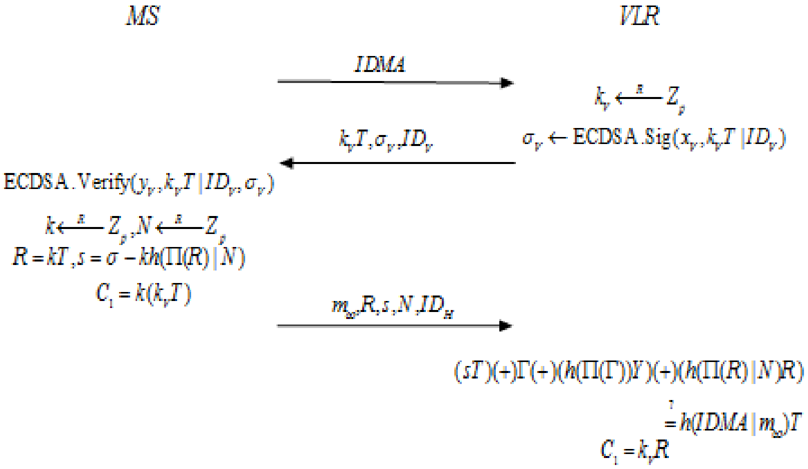

MS
VLR
$IDMA$
$k_v \xleftarrow{R} Z_p$
k_vT, σ_v, ID_v
$\sigma_v \leftarrow \text{ECDSA.Sig}(x_v, k_vT \mid ID_v)$
$\text{ECDSA.Verify}(y_v, k_vT \mid ID_v, \sigma_v)$
$k \xleftarrow{R} Z_p, N \xleftarrow{R} Z_p$
$R = kT, s = \sigma - kh(\Pi(R) \mid N)$
$C_1 = k(k_vT)$
m_ω, R, s, N, ID_H
$(sT)(+)\Gamma(+)(h(\Pi(\Gamma))Y)(+)(h(\Pi(R) \mid N)R)$
$\overset{?}{=} h(IDMA \mid m_\omega)T$
$C_1 = k_vR$

Figure 4: The protocol based on delegation by warrant

the session key. Otherwise it rejects the connection.

4.2.2 Comparison with the Scheme Based on Group Signature

In our scheme, MS does not send ID_M in plain text but a pseudonym instead. Anyone else including VLR cannot get ID_M. Unfortunately the pseudonym is generated by HLR, so MS cannot change it at will and can be easily traced. On the contrary, the protocol based on group signature in [24] can get strong unlinkability because the pseudonym is given by MS itself and can be changed arbitrarily. Though the unlinkability is weaker, our protocol is more efficient. Only 3.25 ECSM public key operations are needed by MS in our protocol, but 8.75 ECSM plus 3 Pairing operations are needed in [24]. Table 2 is the comparison between them. By using Table 3 from [24], we compare their computation delay in Figure 5, from which we can see that, our protocol needs only one fourth computation delay in [24].

5 Analysis

5.1 Security

In this section, we analyze our proposed scheme in terms of security.

Table 2: Comparison between [24] and our protocol

Schemes	Unlinkability	Public-key computation in MS
[24]	Strong	8.75ECSM+3Pairing
Our protocol	Weak	3.25ECSM

Table 3: Timings on 200MHz processor

	ECSM	Pairing
Time(ms)	23	38

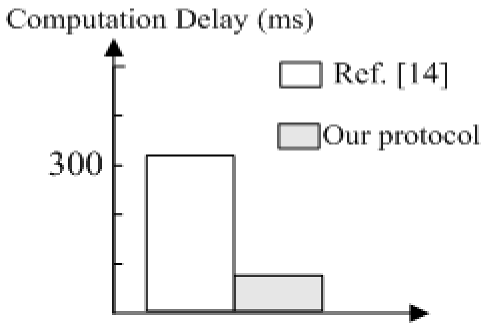


Figure 5: Computation delay in a 200MHz MS

1) Server authentication: In our scheme, MS is sure of the ID of VLR by verifying the signature of VLR.

2) Subscriber validation: MS signs a message on behalf of HLR; VLR verifies it to ensure that MS gets the delegation of HLR and is a valid user.

3) Key establishment: MS and VLR establish a common session key by Diffie-Hellman (DH) key exchange, which cannot be derived by anyone else including HLR.

4) User Anonymity: Besides the user and HLR, anyone else even VLR cannot tell the real identity of MS;

5) Resistance to man-in-the-middle attack: In our second example, an attacker cannot establish a fake Man-in-the-middle session key between MS and VLR because it is impossible for the adversary to get knowledge of the secret key k_v or k. The proposed protocol therefore resists the man-in-the-middle attack.

6) Non-repudiation: In our first example, MS will transmit $h^{n-i+1}(n_1)$ to VLR at the offline authentication phase. The $h^{n-i+1}(n_1)$ is a proof that MS requested VLR's service. Since it's based on hash chain irreversible characteristic, although VLR has the $h^{n_i+2}(n_1)$, which is received from previous communication, it still cannot generate the $h^{n-i+1}(n_1)$ by itself.

5.2 Practicability

In wireless roaming networks such as Cellular Networks, users often roam frequently. When users roam from one visited network to another, re-authentication is inevitable. Too much authentication time will affect the quality of service (QoS), especially in real-time interpersonal communications. Our scheme needs fewer message flows and less computation delay than traditional schemes and the scheme based on group signature respectively. Of course its unlinkability is weaker, which makes it not very satisfactory. But as an option, users can choose it if the bandwidth is not good or their mobile stations are not very powerful.

5.3 Disadvantages

Although our scheme is efficient in real-time interpersonal communications, it still has some disadvantages which may affect its application.

Weaker unlinkability is the first disadvantage which has been discussed above.

The second weakness of our scheme is its complex billing mechanism which is common in two-party protocols without involving HLR. One practical solution is the so-called "D-Coin" billing mechanism which employs the hash-chain technique. This has been discussed and solved in [27].

6 Conclusions

This paper introduces a new kind of delegation-based scheme involving only two parties. It is not only more secure and efficient than these schemes involving three parties, but also more efficient than the scheme based on group signature. Though its unlinkability is weaker, its high efficiency makes it more practical in power-limited and band-limited wireless roaming networks.

Acknowledgements

This work is supported by Science and Technology Project Founded by the Education Commission of Jiangxi Province (No. GJJ161195), Humanities and social science project of universities and colleges of Jiangxi Province (No. JC162001), Science and Technology Project Founded by the Xinyu science and Technology Bureau (No. 20163090862), and the National Nature Science Foundation of China (No. 61402055). The authors would like to thank the anonymous referees for their valuable suggestions.

References

[1] G. R. Alavalapati, A. K. Das, E. J. Yoon, *et al.*, "A secure anonymous authentication protocol for mobile services on elliptic curve cryptography", *IEEE Access*, vol. 4, pp. 4394-4407, 2016.

[2] M. K. Chande, C. C. Lee, C. T. Li, "Message recovery via an efficient multi-proxy signature with self-certified keys," *International Journal of Network Security*, vol. 19, no. 3, pp. 340-346, 2017.

[3] C. C. Chang, C. Y. Lee, Y. C. Chiu, "Enhanced authentication scheme with anonymity for roaming service in global mobility networks", *Computer Communications*, vol. 32, no. 4, pp. 611-618, 2009.

[4] L. Deng, H. Huang, and Y. Qu, "Identity based proxy signature from RSA without pairings," *International*

Journal of Network Security, vol. 19, no. 2, pp. 229-235, 2017.

[5] T. Gao, Q. Wang, X. Wang, *et al.*, "An anonymous access authentication scheme based on proxy ring signature for CPS-WMNs", *Mobile Information Systems*, Article ID 4078521, 11 pages, 2017.

[6] D. He, S. Zeadally, N. Kumar, *et al.*, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures", *IEEE Transactions on Information Forensics & Security*, vol. 11, no. 9, pp. 2052-2064, 2016.

[7] J. S. Kim, K. Jin, "Improved secure anonymous authentication scheme for roaming service in global mobility networks", *International Journal of Security & Its Applications*, vol. 6, no. 3, pp. 45-53, 2012.

[8] P. Kumar, A. Gurtov, J. Iinatti, *et al.*, "Delegation-based robust authentication model for wireless roaming using portable communication devices", *IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, pp. 668-674, 2014.

[9] W. C. Kuo, H. J. Wei, J. C. Cheng, "Enhanced secure authentication scheme with anonymity for roaming in mobility networks", *Information Technology & Control*, vol. 43, no. 2, pp. 151-156, 2014.

[10] C. C. Lee, M. S. Hwang, I. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless environments", *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1683-1687, 2006.

[11] T. Lee, S. Chang, T. Hwang, *et al.*, "Enhanced delegation-based authentication protocol for PCSS", *IEEE Transactions on Wireless Communications*, vol. 8, no. 5, pp. 2166-2171, 2009.

[12] W. Lee, C. Yeh, "A new delegation-based authentication protocol for use in portable communication systems", *IEEE Transactions on Wireless Communications*, vol. 4, no. 1, pp. 57-64, 2005.

[13] L. H. Li, S. F. Tzeng, M. S. Hwang, "Generalization of proxy signature based on discrete logarithms", *Computers & Security*, vol. 22, no. 3, pp. 245–255, 2003.

[14] E. J. L. Lu, M. S. Hwang, and C. J. Huang, "A new proxy signature scheme with revocation", *Applied Mathematics and Computation*, vol. 161, no. 3, PP. 799-806, Feb. 2005.

[15] Y. Lu, X. Wu, X. Yang, "A secure anonymous authentication scheme for wireless communications using smart cards", *International Journal of Network Security*, vol. 17, no. 3, pp. 237-245, 2015.

[16] M. Mambo, K. Usuda, E. Okamoto, "Proxy signature for delegating signing operation", in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pp. 48-57, 1996.

[17] A. Sudarsono, T. Nakanishi, Y. Nogami, *et al.*, "Anonymous IEEE802.1X authentication system using group signatures", *Journal of Information Processing*, vol. 18, pp. 63-76, 2010.

[18] C. Tang, D. Wu, "An efficient mobile authentication scheme for wireless networks", *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1408-1416, 2008.

[19] C. Tang, D. Wu, "Mobile privacy in wireless networks-revisited", *IEEE Transactions on Wireless Communications*, vol. 7, no. 3, pp. 1035-1042, 2008.

[20] S. F. Tzeng, M. S. Hwang, C. Y. Yang, "An improvement of nonrepudiable threshold proxy signature scheme with known signers", *Computers & Security*, vol. 23, no. 2, pp. 174-178, Apr. 2004.

[21] F. Wang, C. C. Chang, C. Lin, S. C. Chang, "Secure and efficient identity-based proxy multi-signature using cubic residues," *International Journal of Network Security*, vol. 18, no. 1, pp. 90-98, 2016.

[22] K. Y. Wu, K. Y. Tsai, T. C. Wu, *et al.*, "Provably secure anonymous authentication scheme for roaming service in global mobility networks", *Journal of Information Science & Engineering*, vol. 31, no. 2, pp. 727-742, 2015.

[23] Y. Xu, L. S. Huang, M. M. Tian, *et al.*, "Insecurity of a certificate-free ad hoc anonymous authentication", *International Journal of Network Security*, vol. 18, no. 5, pp. 993-996, 2016.

[24] G. Yang, Q. Huang, D. Wong, *et al.*, "Universal authentication protocols for anonymous wireless communications", *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 168-174, 2010.

[25] K. H. Yeh, "An anonymous and lightweight autheni-cation scheme for mobile devices", *Information Technology & Control*, vol. 44, no. 2, pp. 206-214, 2015.

[26] H. Zhu, H. Li, W. L. Su, *et al.*, "ID-based wireless authentication scheme with anonymity", *Journal on Communications*, vol. 30, no. 4, pp. 130-136, 2009.

[27] H. Zhu, X. Lin, R. Lu, *et al.*, "Secure localized authentication and billing for wireless mesh networks", in *IEEE Global Telecommunications Conference (GLOBECOM'07)*, pp. 486-491, 2007.

Biography

Chun-lin Jiang received his Ph.D degree in the School of Information Science and Engineering from Central South University, China in 2012. His current research includes security and privacy of next generation wireless communication, protocols and heterogeneous networks.

Shi-Lan Wu received her M.S. degree in Hunan Normal University, Changsha, China, in 2011. She is now a teacher in Xinyu University. Her current research is protocols analysis.

Ke Gu received his Ph.D degree in the School of Information Science and Engineering from Central South University, China, in 2012. His current research includes information security and cryptography.

HKDS: A Hierarchical Key Distribution Scheme for Wireless Ad Hoc Network

Kakali Chatterjee and Lalita Priya

(Corresponding author: Kakali Chatterjee)

Computer Science and Engineering Department, National Institute of Technology
Ashok Rajpath, Mahendru, Patna, Bihar 800005, India
(Email: kakali2008@gmail.com)

(Received June 4, 2016; revised and accepted Sept. 3, 2016 & Feb. 1, 2017)

Abstract

Wireless ad hoc networks are very popular in many areas such as border area protection, rescue operations etc. These networks suffer security problems due to their infrastructure less architecture. Attacks like passive eavesdropping, impersonation, replay etc. can be easily performed in such networks. Also the devices used in such networks are mostly resource constrained devices. Hence highly secured complex cryptographic algorithm cannot implement in such devices. This paper proposes a two level hierarchical key distribution scheme (HKDS) for wireless ad hoc networks. In the first level the secret key is distributed among the cluster heads using knapsack algorithm. In the second level, the secret shares generated by cluster head using Chinese Remainder Theorem (CRT) scheme is distributed among the nodes of the cluster. After that a mutual authentication scheme is introduced through which the node and the cluster head will mutually authenticate themselves and generate the secret session key for communication.

Keywords: Asmuth-bloom Secret Sharing; Ban Logic; Chinese Remainder Theorem; Knapsack Algorithm

1 Introduction

Wireless ad hoc networks are infrastructure less network which provides a basic framework for the ubiquitous computing. It supports anytime and anywhere deployment facility for easy communication. These networks are very disposed to security attacks such as routing attack, node stealing attack, impersonation attack, etc. [14, 20, 27]. Key management plays an important role in such networks as the data need to be encrypted before transmitting to neighboring nodes to resist those attacks. Implementation of suitable key distribution technique in this environment is challenging as the network having limited bandwidth and limited power [19, 21]. Public key management approaches mainly increase computational complexity and communicational overload. Symmetric key distribution techniques are suitable here due to the small key size and simple operations, but the security of the network totally depend upon one key. The network will compromise if the key is hacked. Hence it is better to distribute the secret in such a manner so that the adversary cannot able to get the key even if a node is captured. Different Key management schemes such as centralized group key management scheme [4, 28, 33, 36, 38], contributory group key management scheme [1, 15, 16], hybrid group key management protocol [22, 34], have been proposed in ad hoc networks.

In centralized group key management, key distribution center is involved in distribution of different types of key like group key, temporal key among the group members [8, ?, 11, 31]. Hence these techniques are vulnerable of server spoofing attack and single point failure can occur in the network. To avoid this problem contributory group key management schemes came where contribution of every group member is considered in group key generation process [6, 12, 24]. However these schemes also suffer with scalability problem with high computational costs. Hence a hybrid of these two schemes has been proposed which is fault tolerant and also computationally efficient. But these key management schemes also suffers in node compromise attack and overall communicational overhead increased. To avoid the drawbacks, efficient secret sharing technique is preferable. Threshold Cryptography is suitable in such network to distribute the secret shares in the number of nodes. This reduces the chances of vulnerability and redundancy of secret key. Brickell [5] proposed a linear algebra based method which constructs the ideal secret sharing. Along with this he showed how to apply it to find ideal schemes for the multilevel and compartmented access structures. This schema has some scalability problem. Hence Luo *et al.* [25] described a scalable and distributed authentication technique for ad hoc network. They introduce the virtual certificate authority. After that some authentication techniques are found in literature. Zhu *et al.* [39] first introduced a threshold cryptography based key management system for ad hoc network.

In their work, they define a group of N servers together with a pair of master public-private key which would be deployed by Certificate Authority. Each server was sharing the master private key and stores key pair of all nodes. But due to the lack of a prior knowledge of post-deployment configuration, when, N number of servers come together, they were not able to form a whole signature [26]. Condition for any node who wants to join the network was that they must collect all the N partial signatures from other nodes and compute the whole signature. Ma *et al.* [26] discussed the use of threshold cryptography in opportunistic network. They proposed identity based cryptography (IBC) security scheme where the nodes have to encounter t out of n public key generators to reconstruct their private key. Zou *et al.* [41] proposed an approach for weighted multi secret sharing scheme. Hui et.al [40] proposed a novel group key management scheme for mobile ad-hoc network where registration center handles complete registration of members and panel of key generator center handles key management. After the registration process, the user gets the shared key which is given by panel of key generation center. Farras *et al.* [9] proposed a work for constructing hierarchical secret sharing and the characterization of ideal access structures. Wang *et al.* [36] proposed a secret sharing scheme that distributes its share to currently available member of networks and threshold member will combines it to issue signature. Gharib *et al.* [10] proposed KERBEROS in mobile ad-hoc network. They assumed a predefined trusted third party. In their system, a mobile node sends resource ticket and authenticator to the service provider encrypted with the key. Wang *et al.* [37] proposed an identity based group key communication scheme based on bilinear pairing. In general all these approaches either need higher configuration effort before deployment or higher energy consumption for large traffic generation. Also most of the schemes cannot resist the major vulnerable attacks in ad hoc networks such as node-compromise attack, flooding attack, replay attack etc.

In this paper, we have proposed a two level hierarchical key distribution scheme (HKDS) for wireless ad hoc networks. The root node is the base station (BS). In the first level the secret key is distributed among the cluster heads (CH) using Knapsack algorithm. In the second level, the secret shares generated by cluster head using Chinese Remainder Theorem (CRT) scheme is distributed among the nodes of the cluster. After that in mutual authentication phase the node and the Cluster Head will mutually authenticate themselves and generate the secret session key for communication. We have compared proposed authentication scheme with some popular authentication protocol [17, 23, 29].

Our proposed scheme also includes the following aspects:

- This scheme efficiently covers two major issues like key management and node authentication in ad hoc network. We use Knapsack algorithm because one major practical advantage over RSA is speed. It can operates at throughput rates of 20 mbits/sec whereas in RSA through put rate is about 50 kbits/sec. Knapsack is suitable for resource constrained environment because it avoids complex operations like modular exponentiations.
- In first level, Knapsack key is used for communication between BS and CH which provide the strength of public key cryptography in this level. In second level, Secret shares are generated from the knapsack key and stored in cluster nodes. From the shared secret key the session key is generated which is used for message encryption (using AES) between cluster nodes and cluster head. Symmetric key cryptography is used in this level for reducing computational complexity.
- This scheme checks node validity and mutual authentication between the node and the CH. After that they will generate the secret session key.
- Authentication approach enforces very light computational load and detail security analysis shows that it resists all possible attacks.
- To resist node compromise attack, secret shares are generated and distributed in n number of nodes (participants). For recombination of secret need minimum t number of nodes (participants). Hence if a node is captured, then also an advisory cannot compute the secret key.
- This scheme uses AES symmetric encryption process to save energy and storage which is critical for constrained devices.

The rest of the paper is organized as follows: Section 2 presents Backgrounds, Section 3 provides Proposed Authentication Protocol; Section 4 discusses Implementation results; Section 5 presents Security Analysis; Finally, we conclude the paper in Section 6.

2 Backgrounds

In this section, we will discuss challenges of wireless ad hoc networks and threshold based cryptosystems.

2.1 Security Challenges in Ad Hoc Network

Ad hoc networks are decentralized and dynamic in nature [35]. The basic challenges for ad hoc network are:

- In ad hoc networks the packet of data is very insecure due to hostile environment.
- Each device contains low end processor having less speed and small programmable memory.

- Public key cryptosystems degrade the performance for complex mathematical operations.
- Physical capture of node by the adversary is a common problem.
- Due to dynamic nature, there is lack of post-deployment configuration knowledge.
- The nodes communicate each other and the base stations using low bandwidth and less transmission power.
- Group key management leads more communicational overhead.
- Ad hoc network devices are generally operated by batteries. Battery technology is lagging behind microprocessor technology. The life time of a battery have lower time range which implies the need of power conservation.
- Routing attacks (sinkhole, black hole), selective forwarding, node tampering, jamming and flooding attacks are possible attacks in these networks.

2.2 Threshold Based Cryptosystem

Secret sharing is a method in which we distribute the shares of the secret to the share-holders. The secret will be recovered only by certain predetermined groups as per access structure definition. The secret sharing schemes, where only a limited number (threshold) of participants in the reconstruction phase is important for recovering the secret is called threshold secret sharing scheme. When it calculates for total weight as threshold, it is named as weighted threshold secret sharing. Secret sharing scheme usually can be divided into following steps:

Dealer phase: This scheme starts from this phase, as dealer coordinates the whole share distribution scheme. Dealers generate a secret and its shares and distribute them in participants.

Combiner Phase: Combiner can be a participant or a special party that collect the shares from authorized participants and regenerate the secret.

For example, if a secret S_0 has to be distributed in n number of participants and threshold defined by access structure is t then dealer will generate n number of share counting from $S_1, S_2, \cdots, S_n$. Combiner will collect any t number of shares and recalculate the value S_0. According to the availability of secret to the dealer, the secret sharing scheme is defined as:

Explicit: Dealer receive the secret from outside and generate shares on it.

Implicit: Dealer create or have the secret and generate the shares on it. Generally dealer generates the secret from predetermined domain. Two types of secret sharing scheme is found in literature given below:

1) Shamir secret sharing scheme [32]: Shamir secret sharing scheme was based on polynomial interpolation. It's equation is in the form of any K-pairs $(X_1, Y_1), (X_2, Y_2), \cdots, (X_k, Y_k)$ with $x_i \neq x_j$. Dealer give a polynomial equation in the form of $p(x)$ degree $(t-1)$ such that $p(x_i) = y_i$ for all $1 \leq i \leq k$. Some features of Shamir secret sharing are:
    - Secret is chosen as free coefficient of a random polynomial.
    - Share is chosen as $l_i = P(x_i)$ for all $1 \leq i \leq n$ with x_i as a different public value.
    - Secret is recombined by using Lagrange's interpolation.
2) Blakey's Secret Sharing Scheme [3]: Blakey used n-dimensional vector space. It presents the secret as an element of GF_q^k vector space. Share were taken as any n-different $(t-1)$ dimensional vector space. Share were taken as any n different $(t-1)$ dimensional hyper plane subset of dimensional vector space as,

$$\begin{aligned} a_{11}x_{11} + a_{12}x_{12} + \cdots + a_{1t}x_{1t} &= a_1 \\ a_{21}x_{21} + a_{22}x_{22} + \cdots + a_{2t}x_{2t} &= a_2 \\ \vdots \quad & \vdots \quad \vdots \\ a_{n1}x_{n1} + a_{n2}x_{n2} + \cdots + a_{nt}x_{nt} &= a_n. \end{aligned}$$

Secret get recovered by intersection of K shares. The secret sharing scheme based on Chinese Remainder theorem has been proposed in [13]. It is a method to uniquely determine a number S modulo k many relatively prime integers $m_1, m_2, \cdots, m_k$, given that $S < \prod_{i=1}^{k} m_i$. The shares are generated by reduction modulo the integer m_i, and the secret is recovered by essentially using the Chinese remainder theorem.

3 The Proposed Authentication Protocol

In this section we propose a hierarchical authentication protocol for ad hoc network. Network model is given below.

3.1 System Network Model

Consider a system network model for border area protection. The ad hoc network is deployed for collecting the data regarding any motion or disturbances created in border area. The whole area is divided into small clusters and each cluster has a cluster head. The message from the nodes will transmit to the root node in an encrypted form [7]. During this process, each node authenticates itself to the cluster head before transferring data. Hence we design the two level hierarchical models for key distribution and authentication. In this hierarchical model, let there are number of clusters and each cluster head will separately calculate and distribute its key among their cluster node. In deployment model, Level 1 connection (from root node to cluster head) is infrastructure based and Level 2 connection (from cluster head to cluster nodes) is infrastructure less. The network structure is described in Figure 1.

The proposed hierarchical key distribution scheme is based on Knapsack Cryptosystem and CRT based secret sharing scheme. First level is for key distribution of root node to cluster head using knapsack method and second level is for key distribution of cluster head to cluster node using CRT based secret sharing. In this process the generated symmetric key is used for node to node communication. While the node and cluster head will establish a session key for further data communication.

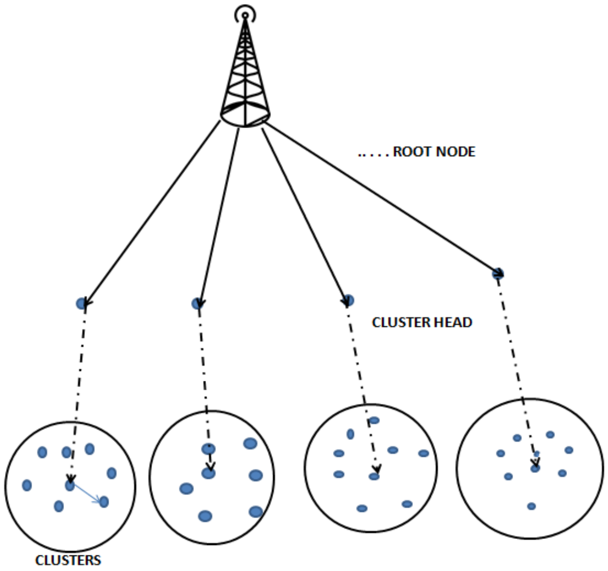


Figure 1: System model of HKDS

Assumptions used in this protocol:

- In this model, the root node is a trusted node. Mainly the Base Station is assumed here the root node.
- Cluster nodes are the mobile devices which are under the control of Cluster Heads.
- Cluster head, cluster nodes and root node are considered static and synchronized.
- In this system model, there is number overlapping between clusters.
- The communication between cluster head to cluster node is insecure.
- Threshold is calculated by static method.
- Signal intensity of all nodes is considered under threshold value.

Table 1 shows the notations used in this protocol.

Table 1: Notations used in this protocol

Symbols	Meaning
UN	User node
Bs	Base station/root node
CH	Cluster head
UID	Identification number of user node
CID	Identification number of cluster head
P_u	Random number chosen by user
R	Registration number
$h()$	Hash function (SHA-1 hash algorithm is used)
g	A generator on Z_p* where $(2 \leq g \leq p-2)$
P_b	One time key between root node and cluster node
P_{ch}	One time key between root node and cluster head
P_m	One time master key between cluster node and cluster head
P_k	Knapsack tuple
A	Diffie Hellman key of user
B	Diffie Hellman key of cluster
M_i	i^{th} message
T_i	i^{th} time stamp
N_i	i^{th} share of nodes
Key	Symmetric key in ad hoc network calculated by user node
K	Temporary session key
K_{SN}	Final session key

3.2 Description of Proposed Authentication Protocol

Proposed authentication protocol is divided in two phase. First phase is key distribution and registration which will be performed after deployment. Second phase is for Login and Authentication of nodes for data exchange.

Phase I (Key Distribution and Registration).

1) Key distribution Phase: This key distribution phase works in two level as discussed in the previous work [30]. In first level key distribution is performed from root node to cluster head using knapsack algorithm and in second level cluster head calculate secret key for each node using CRT based secret sharing and calculates its corresponding triplet,then send it to the cluster node. The key distribution is shown in Figure 2.

Level 1 (Root node to cluster head).
Root node generates Knapsack keys using N tuples super increasing key series where N is equals to the number of cluster head to which the key is to be distributed. Each tuple of the series will be given to a single cluster head. This tuple is the key for cluster head node and its share will be distributed among the cluster nodes. The process is given below: For N clus-

ter head, we choose super increasing series of N natural number.

$$w = (w_1, w_2, \cdots, w_N).$$

Randomly select a integer q such that

$$q > \sum_{i=1}^{N} w_i \qquad (1)$$

and selects r such that $1 \leq r \leq q-1$ and $\gcd(q,r) = 1$ and now calculate

$$\beta_i = rw_i \bmod q. \qquad (2)$$

So the calculated series

$$\beta = (\beta_1, \beta_2, \cdots, \beta_n).$$

Permute the β series and find new series

$$\gamma = (\gamma_1, \gamma_2, \cdots, \gamma_n).$$

This γ series will be the public key and the series

$$w = (w_1, w_2, \cdots, w_n), q, r,$$

will be the private key of root node. The public key tuples will be distributed to the cluster heads. For reconstructing the public key, the entire cluster node will exchange their share with their identity. Thus here the threshold $t = n$. Similar way CH will generate public-private key and send to root node. Finally root node will create a public key list of all CH.

Level 2 (Cluster head to cluster node connection).

In this level, CH will distribute the key to the node using CRT based secret sharing scheme. A special sequence of integer used here is known as Asmuth-bloom sequence [2];

$$p_0, p_1 < p_2 < \cdots < p_n.$$

Here n is the number of nodes in a cluster and threshold is decided at t. This sequence must satisfy the equation

$$A_0 \prod_{i=0}^{t-2} A_{n-i} < \prod_{i=1}^{t} A_i. \qquad (3)$$

The dealer phase and combiner phase will separately run in each cluster by its cluster head.

Dealer phase.

p_0 is selected as the secret S belongs to element of z_{p_0}. The cluster node select a random number α so that

$$p_{n-t+2} \times p_{n-t+3} \times \cdots p_n < S + \alpha p_0 < p_1 \times p_2 \times \cdots \times p_t.$$

This value α will determine that without participation of nodes the secret key of cluster head cannot be retrieved. Secondly if value of $S + \alpha p_0$ is lower than the lower range decided for threshold can be reconstructed by combining less than threshold number of shares. Shares can be calculated by:

$$s_i = (S + \alpha p_0) \bmod p_i. \qquad (4)$$

Combiner phase.

Cluster head will collect the threshold no of shares and calculate according to Chinese remainder theorem.

$$\begin{aligned} x &= s_1 \bmod p_1 \\ x &= s_2 \bmod p_2 \\ &\vdots \\ x &= s_t \bmod p_t. \end{aligned}$$

Here $Z = p_{i_1} p_{i_2} \cdots p_{i_t}$ and value of x can be calculated by $x = \sum_{i=1}^{t} \frac{z}{A_{i-1}} y_i s_i \bmod Z$. After calculating X, we can calculate the secret that is a key for each cluster node as $S = x \bmod p_t$.

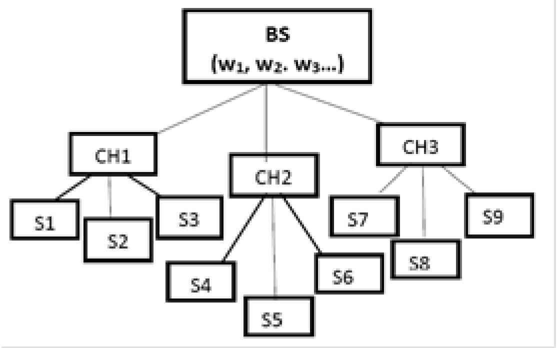


Figure 2: The Key distribution structure in HKDS

Now the cluster-head will calculate share for each node but it will send other factors along with the shared key to calculate the secret key. The steps involved in this process are given below:

Step 1: After calculating shared key it will find total multiple.

$$M = s_1 \times s_2 \times \cdots \times s_n \times p_{n+1}.$$

p_{n+1} is prime number which is greater than p_n.

$$M_1 = s_1 \times s_2 \times \cdots \times s_n.$$

Step 2: Calculate total sum

$$\begin{aligned} Sum &= s_1 + s_2 + \cdots + s_n + p_{n+1} \\ Sum_1 &= s_1 + s_2 + \cdots + s_n. \end{aligned}$$

Table 2: Message flow in registration phase

Message Flow	Message Name	Message Description
$UN \rightarrow BS$	Sub_{id}	$UID\|\|PIN\|\|P_u$
$BS \rightarrow UN$	Reg_{no}	$E_{P_b}(h(PIN)\|\|R)$
$BS \rightarrow CH$	$Node_{list}$	$E_{P_{ch}}(UID\|\|h(PIN)\|\|Q\|\|P_k)$
$CH \rightarrow UN$	$Node_{info}$	$E_{P_m}(secretshares)$

Step 3: Calculate symmetric key for cluster node as

$$Secretkey = M \bmod Sum.$$

Symmetric key will be stored as (M_1, Sum_1, p_{n+1}) Instead of sharing the symmetric key directly, cluster head will distribute this key in the form of triplet. Cluster-head will calculate triplet for each share applying conditions as follows:

Condition 1: If shared key = 1 or 0.

Step 1: Calculate array of prime number greater than total multiple . Size of array will be the total number of 0 and 1. Then calculate, multiplicative inverse of total multiplication.

$$M_1 = (M, Z_{[p_x]}*).$$

Step 2: Calculate Multiplicative inverse of Sum

$$ISum = (Sum, Z_{[p_x]}*).$$

Step 3: Calculate Triplet $(M_1, ISum, Sharedkey)$ to be send to node.

Condition 2: If $sharedkey > 1$

Step 1: Calculate $M_1 = \frac{M}{SharedKey}$ and $ISum = Sum - SharedKey$.

Step 2: Calculate Triplet $(M_1, ISum, SharedKey)$ to be send to node. After receiving the triplets, node will calculate the symmetric key by applying conditions as follows.

Condition 1: When shared secret is 1 or 0, calculate:

$$\begin{aligned} M &= inverse(M_1, Z_{[p_x]}*) \\ Sum &= inverse(ISum, Z_{[p_x]}*) \\ SecretKey &= M \bmod Sum. \end{aligned}$$

Condition 2: When $SecretKey > 1$, calculate:

$$\begin{aligned} M &= M_1 \times SharedSecret \\ Sum &= Sum_1 - SharedSecret \\ SecretKey &= M \bmod Sum. \end{aligned}$$

2) Registration Phase.
After deployment the nodes will registered to the base station by using following steps:

Step 1: The node will submit its UID and PIN with a random number p_u in registration form and submit it to base station. UID is a 6 digit hexadecimal number and PIN is a 4 digit number. UID is fixed (never be changed), but user can change his PIN when it is compromised. Base station will calculate $R = H(UID||p_u)$ and $Q = R \oplus H(PIN)$.

Step 2: BS will send hash of PIN concatenated with R, g to user node using the direct link. Both BS and User node will store hashed form of PIN.

Step 3: BS will send node-list UID, PIN, Q, and knapsack tuple key to cluster head. Encrypted by one time predefined key between BS and cluster head. When the registered cluster-node deployed to CH than CH will verify its UID and send it the shared triplet. Table 2. shows the message flow in this phase. This table shows sequence of messages in registration phase and what message parameter is passing from one node to another node.

Phase II (Login and Mutual Authentication Phase).
This phase discuss login process, session key generation and mutual authentication process. After authentication, session key is established. The mutual authentication phase is shown in Figure 3.

Step 1: During login process, user node submits it's UID and R to cluster head. The CH than calculate $Q' = R \oplus H(PIN)$ as CH already have $H(PIN)$ of the user node in user list table. If $Q' = Q$, where Q is already stored in the table, than login permitted and send $Grant_{login}$ message to user node. This message contains a nonce $h(N_1)$.

Step 2: Now user node calculates $M_1 = (R||h(UID)||h(N_1))$, $M_2 = (M_1||T_1)$, $A = g^a \bmod p$, $M_3 = h(A||M_1)$ and send $[CID, UID, M_2, A]_{E_{Key}}$ to cluster head.

Step 3: Cluster head verify the freshness of message by $T_2 - T_1 = \delta T$ and generate $M_1' = (R||h(UID)||h(N_1))$

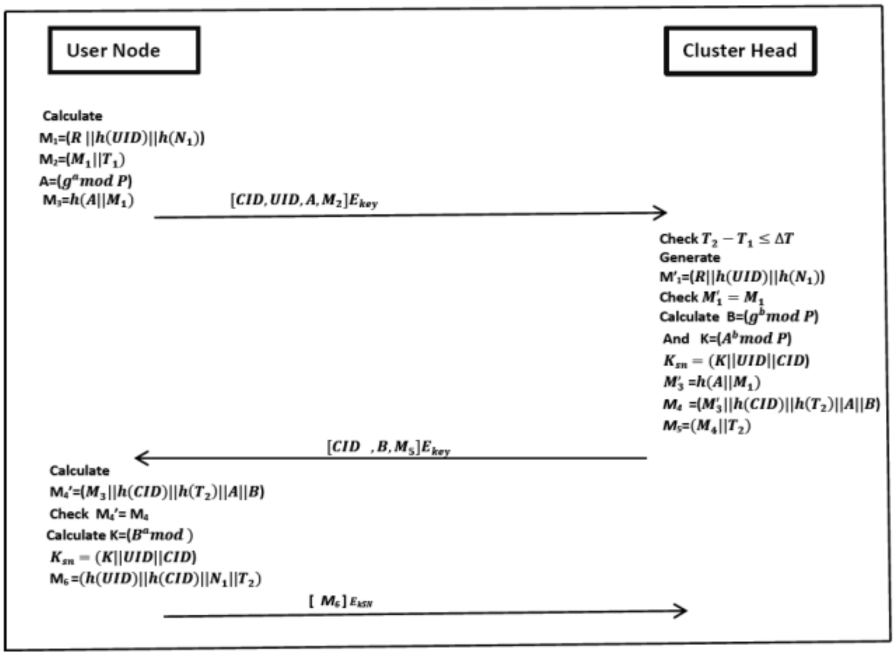


Figure 3: Mutual authentication phase

and verify it with M_1. Cluster head consider a random number b and calculate B by $g^b \bmod p$. Cluster head extract A from the message and generate $K = A^b \bmod p$. It calculates $M_2' = h(A||M_1)$ and the session key by $(K||UID||CID)$. Cluster head append M_3' in M_4. Calculate $M_4 = (M_3'||h(CID)||h(T_2)||A||B)$ and $M_5 = (M_4||T_2)$. Cluster head send message M_5 to user node as $[CID, B, M_5]_{E_{Key}}$.

Step 4: User node receive M_4 and N_2 User node calculate $M_4' = (M_2||h(CID)||h(T_2)||A||B)$. It checks $M_4' = M_4$ and calculates $K = B^a \bmod p$ and session key $K_{SN} = (K||UID||CID)$. User node send response message $M_6 = [(h(UID)||h(CID)||N_1||T_2)]_{E_{Key}}$.

Table 3 shows the message flow in this phase. In this way mutual authentication is performed.

4 Implementation and Performance Analysis

Implementation of the first phase is performed in Java Platform using Java Cryptographic Extension. We use a Laptop with Intel Core 2 DUO CPU T6400@2.00GHz with 4GB RAM and Windows 7 operating system having jdk1.8 as a Cluster head node. Implementation is distributed in three phases:- Knapsack Key generation phase, Share generation phase, and lastly the triplet generation phase. As an example, consider a cluster with 9 nodes and the share generated for 9 nodes is shown in Figure 4. Here a 15 digit random number is chosen and it is equal to 782490775074582. With this 9 generated shares, the cluster head will calculate triplet to send the nodes as shown in Figure 4 (screenshot) below using Netbeans IDE 6.8. Now the nodes can calculate the secret session key for encryption of message.

Next we calculate how much time is consumed for encrypting messages with this shared key in a mobile device (a tablet with 2GB RAM, ARM Cortex 1.7 GHz processor). For a Text File (File size: 870 KB) the encryption time is 30 ms and decryption time is 28 ms for same size data using the shared key.

To calculate how much energy it consumes over time, we find that the half of battery energy of the mobile device (a tablet with 2GB RAM, ARM Cortex 1.7 GHz processor with 3500 mAH polymer battery) consumes after 2 hours while using AES with fixed input block of length 128 byte and different key sizes (128 bytes,192 bytes, 256 bytes) for data transfer.

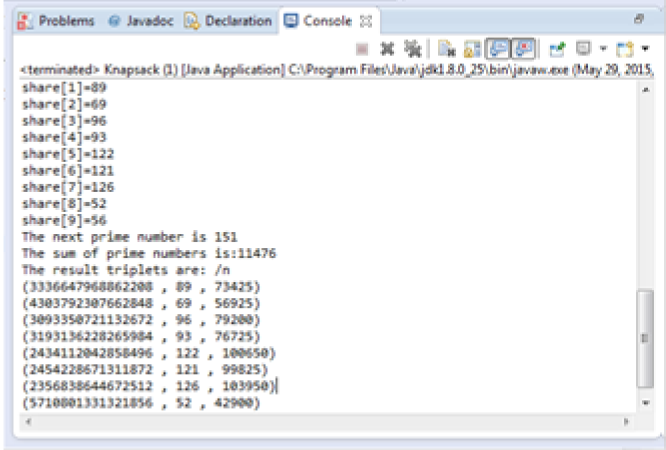
While calculating computation costs, we use the notation TH as the time complexity for the hashing function and TE as the time complexity for exponentiation function and TM as the time complexity for modular multiplication function. In cryptography, Exclusion-OR operations are usually neglected (due to very low computations) while considering its computational cost. Communicational cost depends upon the total no. of message exchanged for authentication. Here only three messages exchanged in mutual authentication phase. The comparison of our proposed protocol with other protocols is given in Table 4. Also functionality comparison is given in Table 5.

Table 3: Message flow in login and mutual authentication phase

Message Flow	Message Name	Message Description
$UN \rightarrow CH$	Req_{Login}	$UID\|\|R$
$CH \rightarrow UN$	$Grant_{login}$	$h(N_1)$
$UN \rightarrow CH$	$User_{Authn}$	$E_{key}(CID\|\|UID\|\|M_2\|\|A)$
$CH \rightarrow UN$	CH_{Authn}	$E_{key}(CID\|\|B\|\|M_5)$
$UN \rightarrow CH$	$Sesn_{Gen}$	$E_{key}(h(UID)\|\|h(CID)\|\|N_1\|\|T_2)$

Table 4: Performance comparison of our scheme

Schemes	Login phase	Authentication phase	Total
Lee *et al.* [17]	$7T_H$	$9T_H$	$16T_H$
Pippal *et al.* [29]	$2T_E + T_M + T_H$	$5T_E + T_M + 6T_H$	$7T_E + 2T_M + 6T_H$
Li *et al.* [23]	$T_E + 5T_H$	$3T_E + 8T_H$	$4T_E + 13T_H$
Our Scheme	T_H	$4T_E + 10T_H$	$4T_E + 11T_H$



```
<terminated> Knapsack (1) [Java Application] C:\Program Files\Java\jdk1.8.0_25\bin\javaw.exe (May 29, 2015,
share[1]=89
share[2]=69
share[3]=96
share[4]=93
share[5]=122
share[6]=121
share[7]=126
share[8]=52
share[9]=56
The next prime number is 151
The sum of prime numbers is:11476
The result triplets are: /n
(3336647968862208 , 89 , 73425)
(4303792307662848 , 69 , 56925)
(3093350721132672 , 96 , 79200)
(3193136228265984 , 93 , 76725)
(2434112042858496 , 122 , 100650)
(2454228671311872 , 121 , 99825)
(2356838644672512 , 126 , 103950)
(5710801331321856 , 52 , 42900)
```

Figure 4: Triplet generation

5 Detail Security Analysis

In this section we discuss the security of proposed scheme. BAN logic is used here to proof the mutual authentication between the node and the cluster head and shared a session key. Then, we describe how the proposed protocol resists other network attacks.

5.1 Security Proof Using BAN Logic

We have proved that the authentication protocol provide high security using BAN Logic. BAN Logic is the defined set of logical rules for verifying the correctness of any protocol [23]. It also defines the beliefs of participants in the communication. Correctness of a protocol defines that both communicational parties confirms that they are sharing a fresh session key with each other after execution of the protocol.

For security verification, this work first starts with its normal definition found in [18]:

- R and S are principals i.e. the participants which communicate.
- I and J are statements.
- Key is the cryptographic keys.

Relationships and its uses for all principal, key and statement:

- $\#(I)$: The formula I is fresh.
- $R \models I$: R believes that I is true.
- $R \rightarrow I$: R is an authority and believes I.
- $R\delta I$: R receives some message including I from someone.
- $R \Vdash I$: R sent a message containing I sometime.
- (I, J): The formula I or J is one part of the formula $I, J)$.
- $< I > I$: The formula I combines with a secret parameter J.
- $\{I\}_{Key}$: The formula I is encrypted with the key Key.
- $(I)_h$: The formula I is hashed.
- R_Key_S: R and S use the shared key Key to communicate and Key will never be discovered by any principal except R and S.
- Message meaning rule: $\frac{R \models R\_Key\_S, R\delta\{I\}_{Key}}{R \models S \Vdash I}$.
- Freshness conjugation rule: $\frac{R \models \#(I)}{R \models \#(I,J)}$.

- Freshness-introduction rule: $\frac{R\ creat\ a\ random\ I}{R\models\#(I)}$.
- The belief rule: If the principal R believes I and J, then the principal B believes (I, J): $\frac{R\models I\models J}{R\models(I,J)}$.
- The nonce-verification rule: If the principal R believes that I is fresh and the principal S sent I once then the principal R believes that S believes I: $\frac{R\models\#(I), R\models S\Vdash I}{R\models S\models I}$.
- The jurisdiction rule: If the principal R believes that S has jurisdiction over I and S believes I, then R believes that I is true: $\frac{R\models S\rightarrow I\models S\models I}{R\models I}$.
- Introduction of the session keys: If the principal R believes that the session key Key is fresh and the principal S believes I. This is essential for a key, then R believes that he/she shares the session key Key with S: $\frac{R\models\#(S), R\models S\models I}{R\models R\_Key\_S}$.

For correctness measurement, the key agreement protocol must achieve the following goals:

$$\begin{aligned}
&Goal1 : CH \models CH\_K\_UN\\
&Goal2 : UN \models UN\_K\_CH\\
&Goal3 : UN \models UN\_K_{SN}\_CH\\
&Goal4 : CH \models UN\_K_{SN}\_CH\\
&Goal5 : UN \models CH \models UN\_K_{SN}\_CH\\
&Goal6 : CH \models UN \models UN\_K_{SN}\_CH.
\end{aligned}$$

- Verification of this protocol is as following: This protocol will have three participants: $Root_{node}()$, $Cluster_{head}(CH)$, $User_{node}(UN)$. Verifying this protocol using BAN logic requires some assumption. They are as follows: From the registration phase before deployment they have

  $$\begin{aligned}
  &A1 : BS\_p_B\_UN\\
  &A2 : BS\_p_N\_CH\\
  &A3 : CH\_p_M\_UN\\
  &A4 : BS\_K_{NC}\_UN, UN\_K_{NC}\_UN\\
  &A5 : CH \models UID\\
  &A6 : UN \models UID\\
  &A7 : CH \models CID\\
  &A8 : UN \models CID.
  \end{aligned}$$

  Verification of this protocol using BAN logic follows: Starting from the First message $C_1UN \rightarrow CH[CID, UID, A, M_2]_{K_{NC}}$. A is random variable calculated by Diffie Hellman process. Thus, we can assume that $A9 : UN \models \#(A)$. M_2 is calclated at UN as $M_2 = [M_1||T_1]$. Whereas, $M_1 = [R||h(UID)||h(N_1)]$. So again we can conclude that $A10 : UN \models \#(M_2)]$ and $S1 : CH\delta[CID, UID, A, M_2]_{K_{NC}}$. CH has seen the message. By message meaning rule:

  $$S2 : CH \models UN \Vdash [CID, UID, A, M_2]_{K_{NC}}. \tag{5}$$

By verifying $T_2 - T_1 = \delta T$, $S3 : CH \models \#(T_1)$. So, By Freshness conjugation rule:

$$S4 : CH \models \#(A). \tag{6}$$

By Freshness conjugation rule: $S5 : CH \models \#(M_1)$. From Equations (5) and (6),

$$S6 : CH \models UN \models (A). \tag{7}$$

Now, CH will calculate K with the help of A. So, we can conclude.

$$A11 : CH \models \#(K). \tag{8}$$

From Equations (7) and (8) and by introduction of session rule, we get $S4 : CH \models CH\_K\_UN$. Goal 1 is achieved.

Again, for Message 2, $C2 : CH \rightarrow UN[CID, B, M_5]_{K_{NC}}$. B is random variable calculated by Diffie Hellman process. Thus, we can assume that $A12 : CH \models \#(B)$. M_5 is calculated at UN as $M_5 = [M_4||T_2]$, whereas $M_4 = [M_2'||h(CID)||h(T_2)||A||B]$ and $S7 : UN||\delta[M_2'||B||CID||T_2]_{K_{NC}}$. UN has seen the message. By message meaning rule:

$$S8 : UN \models CH \Vdash [M_3'||B||CID||T_2]_{K_{NC}}. \tag{9}$$

By verifying freshness of T_2 and M_5. We get

$$S9 : UN \models \#(B). \tag{10}$$

From Equations (9) and (10), we get

$$S9 : UN \models CH \models \#(B). \tag{11}$$

Since UN also calculate K thus, it can be assumed that

$$A12 : UN \models \#(K). \tag{12}$$

From Equations (11) and (12) and by introduction of session key rule, we get $S9 : UN \models UN\_K\_CH$. Goal 2 is thus achieved.

Now since, K is the temporary session key so both share this K. Thus, we can make assumption that:

$$A13 : UN \models K$$

$$A14 : CH \models K$$

$$A15 : UN \models CH \models K \tag{13}$$

$$A16 : UN \models CH \models K. \tag{14}$$

Both end CH and UN will calculate Session key K_{SN}. Our aim is to establish session key between user node and cluster head, thus

$$A17 : UN \models \#(K_{SN}) \tag{15}$$

$$A18 : CH \models \#(K_{SN}). \tag{16}$$

From Equations (13), (refe10), (15), (16) and using session key rule: $S10 : UN \models UN\_K_{SN}\_CH$. Goal 3 is thus achieved.

$S11 : CH \models UN\_K_{SN}\_CH$. Goal 4 is achieved.

Similarly from freshness rule and Equations (13), (refe10), (15), (16): $S12 : UN \models CH \models UN\_K_{SN}\_CH$. Goal 5 is thus achieved.

$S13 : CH \models UN \models UN\_K_{SN}\_CH$. Goal 6 is achieved.

Hence according to $S4$, $S9$, $S10$, $S11$, $S12$, $S13$, the proposed protocol achieves all the Goals and both user node and cluster head believe they share a session key K_{SN}.

5.2 Security Proof Using Attack Analysis

In this section we discuss the security of proposed scheme. The proposed protocol will be considered to be a secure authentication protocol, if it satisfies the following properties:

Man-in-the-middle attack: In this attack, the attacker establishes a common key between two parties and intercepts all message transmitted between them [7]. He modifies these intercepted messages within a valid time period. This protocol establishes a secret session key K_{SN} without revealing any information about the session key. The share key K of each node is transmitted in triplet form to the nodes. When this share is given to cluster nodes, each node calculates its secret key using an in built algorithm and communicates with other nodes. If adversary comes to know any random triplet of share, then also attacker will be unable to get the share of other node. Without knowing the algorithm he cannot deduce the symmetric key used for communication in cluster. Hence this attack cannot be successful.

Impersonation attack: This attack happens when an attacker impersonates as legitimate user by supplying valid credentials in login process. During login process, user node submits it's UID and R to cluster head. The CH than calculates $Q' = R \oplus H(PIN)$ as CH already have $H(PIN)$ of the user node in user list. If $Q' = Q$, where Q is already stored in the table, than login permitted. Now suppose, the attacker get the information about UID and R. He impersonates as a valid user and establishes a connection. But the shared key is unknown to him. So he will use different key while sending the message $[CID, UID, M_2, A]_{E_{Key}}$ to cluster head. The CH will immediately reject the message for using wrong key. Hence the proposed scheme can resist this type of attack.

Stolen-verifier attack: In this attack, the attacker may be able to steal the verification list from server. In this proposed scheme $H(PIN)$is stored in the verification table. If an attacker steal the verifier $H(PIN)$from the table, then also he will unable to calculate $R = H(UID||p_u)$ as p_u is unknown. So $Q = R \oplus H(PIN)$ is also impossible to compute. Hence this attack will be unsuccessful for determining valid login message.

Replay attack: Our protocol protects replay attack as it depends upon timestamp values (T_1, T_2). Also it depends upon random numbers a, b to confirm the freshness of the request message $[CID, UID, M_2, A]$ and response message $[CID, B, M_5]$. Even if an attacker intercept the request message, then also he will unable to compute M_6 and the correct key k for encrypt the message. It is impossible to compute a from $A = g^a \bmod p$ as it lies on discrete logarithm problem. Thus this protocol resists replay attack.

Perfect forward secrecy: In our protocol perfect forward secrecy is maintained even if the previous shared key is compromised. The attacker knows the previous shared secret N, but also unable to derive the previous session key $K_{SN} = (K||UID||CID)$ between the user node and cluster head because it had number relation with shared secret. Again suppose the attacker capture the request - response message and try to calculate the temporary session key $K = A^b \bmod p$ from the value of A. But it is impossible as it is based upon the assumption that the discrete logarithm problem is intractable and on the value of b. Thus the property of perfect forward secrecy is satisfied.

Insider attack: The user submits his UID and PIN concatenated with a random number to BS to generate $R = H(UID||p_u)$ and $Q = R \oplus H(PIN)$ which is stored in the memory of user node. During login process, user node submits it's UID and R to cluster head. The CH than calculates $Q' = R \oplus H(PIN)$ as CH already have $H(PIN)$ of the user node in user list table. If $Q' = Q$, where Q is already stored in the table, than login permitted.

Now from the registration message, an insider cannot able to calculate PIN because it is concatenated with a random number p_u called salt. Even the $H(PIN)$ value he can reveal, but that will not help him to generate Q because R cannot be calculated without p_u. Therefore insider attack cannot be possible in this protocol.

Server spoofing attack: This attack is very common in networks where the attacker manipulates the valuable data of legal user by setting up fake server. In order to set up a legal CH, the attacker needs to send the response message $[CID, B, M_5]$. As the request message is an encrypted message, hence to decrypt it the key must be known to the attacker. Suppose the shared symmetric key is known, then also the false CH does not know the share

Table 5: Functionality comparison of our scheme

Attacks	Our Scheme	Lee *et al.* [17]	Pippal *et al.* [29]	Li *et al.* [23]
Man-in-the-middle attack	Yes	Yes	Yes	No
Dictionary attack	Yes	No	Yes	Yes
Node-compromise attack	Yes	Yes	Yes	Yes
Mutual authentication	Yes	No	Yes	Yes
Impersonation attack	Yes	No	No	Yes
Stolen-verifier attack	Yes	Yes	No	No
Insider attack	Yes	Yes	No	Yes
Server spoofing attack	Yes	No	No	Yes
Replay attack	Yes	No	Yes	Yes
Perfect forward secrecy	Yes	No	No	Yes

N_1 for the user node. Hence the attacker cannot compute $M_1 = (R||h(UID)||h(N_1))$, Without M_1, it is impossible to calculate $M_5 = (M_4||T_2)$ because $M_4 = (M_2||h(CID)||h(T_2)||A||B)$ and $M_2 = h(A||M_1)$. Therefore the proposed scheme is secure against server spoofing attack.

Node compromise attack: In this attack, an adversary gets hold of a node physically and gain access of all data, intercept and modify message. In this protocol, the node authentication parameter R is embedded in the software which cannot be extracted. Hence if a node is compromised, the attacker can get the UID but not R. Hence valid login cannot possible. Now suppose the attacker capture the node and use it for valid login. After that he wants to send the request message to CH. But this time also he will unsuccessful for generating $M_1 = (R||h(UID)||h(N_1))$. Also further messages such as M_2, M_3 cannot generated. Hence he will unable to establish a session key between cluster head and user node. Thus proposed protocol resists node compromise attack.

6 Conclusion

A two stage hierarchical key distribution scheme and authentication protocol is proposed in this paper. Key distribution is a combination of knapsack public key cryptography and CRT based secret sharing scheme.

The unique feature of this architecture is that instead of symmetric key, secret shares generated by the cluster head is stored on the nodes. Any outsider cannot get the secret key without knowing n number shares. The threshold value minimum number of cluster node will need to generate share. It works with the symmetric key encryption among the cluster node. In this network, symmetric key is not directly distributed but send in the broken form. So that, if any node capture the message containing key then also capturing node will not be able to calculate the key. Performance analysis of the protocol also shows that our scheme resists the major vulnerable attacks in ad-hoc networks with low computational load.

References

[1] Y. Amir, Y. Kim, C. Nita-Rotaru, *et al.*, "Secure group communication using robust contributory key agreement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 5, pp. 468–480, May 2004.

[2] C. A. Asmuth, J. Bloom, "A modular approach to key safeguarding," *IEEE Transaction on Information Theory*, vol. IT-29, no. 2, pp. 208–210, 1983.

[3] G. Blakley, "Safeguarding cryptographic keys," in *National Conference*, vol. 8, pp. 313–317, AFIPS Press, 1979.

[4] M. S. Bouassida, I. Chrisment, O. Festor, "Group key management in MANETs," *International Journal of Network Security*, vol. 6, no. 1, pp.67–79, Jan. 2008.

[5] E. F. Brickell, "Some ideal secret sharing scheme," in *Advances in cryptology (Eurocrypt'89)*, LNCS, vol. 434, pp. 468–475, Springer, 2001.

[6] T. Y. Chang, W. P. Yang, M. S. Hwang, "Simple authenticated key agreement and protected password change protocol", *Computers & Mathematics with Applications*, vol. 49, pp. 703–714, 2005.

[7] K. Chatterjee, A. De, D. Gupta, "A secure and efficient authentication protocol in wireless sensor network," *Wireless Personal Communication*, vol. 81, no. 1, pp. 17–37, Mar. 2015.

[8] S. M. Chen, C. R. Yang, and M. S. Hwang, "Using a new structure in group key management for Pay-TV," *International Journal of Network Security*, vol. 19, no. 1, pp. 112-117, 2017.

[9] Z. Eslami, M. Noroozi, S. K. Rad, "Provably secure group key exchange protocol in the presence of dishonest insiders," *International Journal of Network Security*, vol. 18, no. 1, pp. 33-42, 2016.

[10] O. Farras, C. Padro, "Ideal hierarchical secret sharing scheme," *IEEE Transaction on Information Theory*, vol. 58, no. 5, pp. 3273–3286, May 2012.

[11] H. Gharib, K. Belloulat, "Authentication architecture using threshold cryptography in Kerberos for mobile ad hoc network," *Advances in Science and Technology Research Journal*, vol. 8, pp. 12–18, June 2014.

[12] P. Hiranvanichakorn, "Provably authenticated group key agreement based on braid groups - The dynamic case," *International Journal of Network Security*, vol. 19, no. 4, pp. 517-527, 2017.

[13] M. S. Hwang, W. P. Yang, "Controlling access in large partially-ordered hierarchies using cryptographic keys", *The Journal of Systems and Software*, vol. 67, no. 2, pp. 99-107, Aug. 2003.

[14] S. Iftene, "General secret sharing based on Chinese remainder theorem," *Electronic Notes in Theoretical Computer Science*, vol. 186, pp. 67–84, 2007.

[15] T. Jeyaprakash, R. Mukesh, "A new trusted routing protocol for vehicular ad hoc networks using trusted metrics," *International Journal of Network Security*, vol. 19, no. 4, pp. 537-545, 2017.

[16] Y. Kim, A. Perrig, G. Tsudik, "Group key agreement efficient in communication," *IEEE Transactions on Computers*, vol. 53, no. 7, pp. 905–921, 2004.

[17] Y. Kim. A. Perrigm, G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *7th ACM Conference on Computer and Communications Security*, pp. 235–24, Nov. 2004.

[18] C. C. Lee, T. H. Lin, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13863–13870, 2011.

[19] J. S. Leu, W. B. Hsieh, "Efficient and secure dynamic ID-based remote user authentication scheme for distributed systems using smart cards," *IET Information Security*, vol. 8, no. 2, pp. 104–113, Mar. 2014.

[20] C. T. Li, M. S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks",*Information Sciences*, vol. 181, no. 23, pp. 5333–5347, Dec. 2011.

[21] C. T. Li, M. S. Hwang and Y. P. Chu, "An efficient sensor-to-sensor authenticated path-key establishment scheme for secure communications in wireless sensor networks", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2107-2124, Aug. 2009.

[22] C. T. Li, M. S. Hwang, C. Y. Liu, "An electronic voting protocol with deniable authentication for mobile Ad hoc networks," *Computer Communications*, vol. 31, no. 10, pp. 2534–2540, June 2008.

[23] D. Li and S. Sampalli, "A hybrid group key management protocol for reliable and authenticated rekeying," *International Journal of Network Security*, vol. 6, no. 3, pp. 270–281, May 2008.

[24] X. Li, J. W. Niu, S. kumara, *et al.*, "An enhancement of a smart card authentication scheme for multi-server architecture," *Wireless personal Communications*, vol. 80, no. 1, pp. 175–192, 2015.

[25] I. C. Lin, M. S. Hwang, C. C. Chang, "A new key assignment scheme for enforcing complicated access control policies in hierarchy", *Future Generation Computer Systems*, vol. 19, no. 4, pp. 457–462, May 2003.

[26] H. Luo, P. Zerfos, J. Kong, *et al.*, "Self-securing ad hoc wireless networks," in *Proceedings of the Seventh International Symposium on Computers and Communications*, vol. 2, pp. 1346–1530, 2002.

[27] Y. Ma, A. Jamalipou, "Opportunistic node authentication in intermittently connected mobile ad hoc networks," in *16th Asia-Pacific Conference on Communications (APCC'10)*, pp. 543–548, 2010.

[28] L. T. Ngoc and V. T. Tu, "Whirlwind: A new method to attack routing protocol in mobile ad hoc network," *International Journal of Network Security*, vol. 19, no. 5, pp. 832-838, 2017.

[29] A. Perrig, D. Song, J. D. Tygar, "ELK: A new protocol for efficient large-groupkey distribution," *IEEE Symposium on Security and Privacy*, pp. 247–262, 2001.

[30] R. S. Pippal, C. D. Jaidhar, S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[31] L. Priya, K. Chatterjee, "A secure authentication scheme in adhoc network using threshold cryptography," in *International Conference on Computing and Communication Technologies (ICCCT'15)*, pp. 152–155, 2015.

[32] R. V. Sampangi, S. Sampalli, "Metamorphic framework for key management and authentication in resource-constrained wireless networks," *International Journal of Network Security*, vol. 19, no. 3, pp. 430-442, 2017.

[33] A. Shamir, "How to share a secret," *Communication of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[34] A. T. Sherman and D. A. Mcgrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444–458, May 2003.

[35] R. Srinivasan, V. Vaidehi, Rajavelu Rajaraman, *et al.*, "Secure group key management scheme for multicast networks," *International Journal Network Security*, vol. 11, no. 1, pp. 33–38, 2010.

[36] S. Tanwar, K. V. Prema, "Threats and security issues in ad hoc network: A Survey Report," *International Journal of Soft Computing and Engineering*, vol. 2, pp. 138–143, Jan. 2013.

[37] D. Wang, J. Teng, "Efficient and distributed authentication scheme for secure communication in MANET," *Journal of Computational Information Systems*, vol. 9, pp. 57–58, 2013.

[38] F. Wang, C. C. Chang, Y. C. Chou, "Group authentication and group key distribution for ad hoc networks," *International Journal of Network Security*, vol. 17, no. 2, pp. 199–207, Mar. 2015.

[39] W. H. Yang and S. P. Shieh, "Secure key agreement for group communications," *International Journal of Network Management*, vol. 11, no. 6, pp. 365–374, 2001.

[40] L. Zhu, Y. Zhang, L. Feng, "Distributed key management in ad hoc network based on mobile agent," in *Second International Symposium on Intelligent Information Technology Application*, pp. 600–605, 2008.

[41] H. Zong, L. Q. Chen, Q. Y. Zhu, "The application of threshold secret sharing in key agreement scheme for MANETs," in *International Conference on Computer Science and Service System*, pp. 837–840, 2012.

[42] X. Zou, F. Maino, E. Bertino, *et al.*, "A new approach to weighted multi-secret sharing," in *International Conference Computer Communication and Network (ICCC'11)*, pp. 1–6, 2011.

Biography

Kakali Chatterjee is an Assistant Professor in Computer Science and Engineering Department of National Institute of Technology Patna, India. She has many published research papers in LNCS (Springer) and reputed International Journals of springer. She is working in the field of Information Security and Cryptography.

Lalita Priya received B.Tech in Computer Science & Engineering from Uttarakhand Technical University, Uttarakhand. She has done her M.Tech from NIT Patna. Her area of research is Cryptography and Network Security. Presently she is working as Associate Software Engineer at CGI.

Design of an AES Device as Device Under Test in a DPA Attack

Septafiansyah Dwi Putra, Ma'muri, Sarwono Sutikno, Yusuf Kurniawan, Adang Suwandi Ahmad

(Corresponding author: Septafiansyah Dwi Putra)

School of Electrical Engineering and Informatics, Bandung Institute of Technology
Jl. Ganesha No.10, Lb. Siliwangi, Coblong, Kota Bandung, Jawa Barat 40132, Indonesia
(Email: sept4.182@s.itb.ac.id)

(Received May 20, 2017; revised and accepted Aug. 27 & Oct. 7, 2017)

Abstract

This paper presents a design for the implementation of the AES encryption algorithm in the hardware system. The proposed device is intended to be a device under test in a differential power analysis (DPA) attack. This device uses AES encryption with 128bit key length and electronic codebook (ECB) mode. The platform used in this device is FPGA- Cyclone IV EP4CE115F29C7. AESAVS is used to test the functionality of the device. This study proposes a design for an AES-128 encryption device synthesized in the Quartus IDE. It will feature support conducting side-channel attacks on real condition.

Keywords: AES128; DPA Attack; Side-Channel Analysis

1 Introduction

Information is a strategic resource that needs to be protected to ensure its safety and security [12]. One method of providing information security is encryption, which is done using a cryptographic algorithm [10, 27]. The decryption of the encrypted information is then done using the key, which is available only to the intended audience of the information. Without the key, decrypting well-encrypted information would be impossible. Any cryptographic algorithm relies on the secrecy of its key [20]. If the secret key is known to parties other than the intended recipient, then the security provided by the cryptographic algorithm would be compromised [6]. Without the key, decrypting well-encrypted information would be impossible.

An attack on an encryption algorithm is an attempt to decrypt encrypted information without the key. Attacks also search for and exploit the weaknesses of the encryption algorithm. Over the past few decades, there have been numerous studies on cryptography and encryption algorithm attacks. These studies resulted in the development of algorithms that provide high data security and authenticity such as RSA, ECC, AES, TDES and DSA [22]. Although the security of cryptographic algorithms on the mathematical level has already been achieved, maintaining the security of cryptographic algorithms in their physical implementation remains a major challenge.

Attacks on cryptographic algorithms initially exploited only the mathematical design weaknesses of algorithms [1]. It was assumed that if a cryptographic algorithm was mathematically secure, then its implementation was also secure. This paradigm changed when Kocher published papers on timing attack [18], and power analysis attack [19]. These papers revealed that an electronic device that implements a mathematically secure cryptographic algorithm can still divulge certain information. This leakage of information, referred to as a side-channel, can be used to find the encryption key. Modern cryptographic algorithms cannot be significantly attacked by using brute force methods and finding mathematical weaknesses in the algorithm [21]. As such, side-channel information such as timing information, power consumption, electromagnetic leaks or even sound/acoustic leaks are additional sources of information that can be used to decrypt the encrypted information. Some cryptanalysis attack techniques require technical knowledge of the internal operation of the cryptographic system that is implemented [2, 13, 16, 29, 30]. But DPA and statistical methods are amongst the most powerful techniques in solving complex cryptographic algorithms [11]. In addition to reviewing the aspects of the attack, researchers have conducted various software and hardware countermeasures against side-channel analysis (SCA). On the software level, these measures include transformation and masking of data [7]. On the hardware level, measures include desychronization and noise generation [25]. These countermeasures generally require high performance and have high costs; as such, they are currently still deemed unsuitable for embedded systems [17, 26].

Currently, there is a pressing need for an electronic communications system that is both fast and secure. One of the fundamentals s in establishing fast and secure communication is the use of cryptographic algorithms in the form of a device or devices. This paper presents the design of an AES encryption device as a device under test (DUT) for differential power analysis attacks to be carried out at a later stage. Prototype hardware was developed to perform AES encryption on an FPGA platform with a 128-bit data width that is compatible with the processor embedded systems. The implementation is optimized in terms of the use of resources at an acceptable rate, so that the target data security can be achieved with limited resources. This study was developed during previous studies [23, 24]. and uses datasets provided in other studies. The use of datasets presents some limitations, especially in the development of countermeasures against attacks. So, there is a need for future studies of this device as a DUT to develop adaptive countermeasures.

2 Related Research

2.1 Comparison of AES Implementation

Implementation of an encryption algorithm in hardware tends to be faster than in software [2]. Reconfigurable hardware is a practical solution for implementing cryptographic algorithms in embedded systems and high-speed applications [3]. In using FPGAs as a computing platform, there are two main goals which are often conflicting: maximizing the efficiency of resources and maximizing processing speed. As such, the precise goal of the particular implementation needs to be identified before implementing the algorithm [4].

AES can be implemented with other encryption algorithms. Its standardized algorithms enable its adoption to other applications in general. AES is a symmetric cipher block algorithm, a type of encryption algorithms with the ability to encrypt data at high speeds without sacrificing the level of security [5]. In addition, AES is more efficient than public key algorithms for encrypting large amounts of data [8, 9]. In the implementation of AES, the absolute confidentiality of the encryption key needs to be maintained to ensure the security of the encrypted information. Implementations of AES must also have excellent performance on different applications and platforms, maintaining such qualities as: high speed, low latency values, low area usage, and low power usage. As such, software and hardware designs for AES are continually being developed.

In this paper, FPGA is chosen as the platform for implementing AES. This choice is based on the information in Table 1 which shows a comparison between three AES implementations: ASIC, FPGA, and software. FPGA has advantages over software implementations based on its capability in parallel processing, pipelining, velocity, and resistance to tampering. Meanwhile, the design process of AES in FPGA, while not as fast as software implementations, is significantly faster than the ASIC implementations.

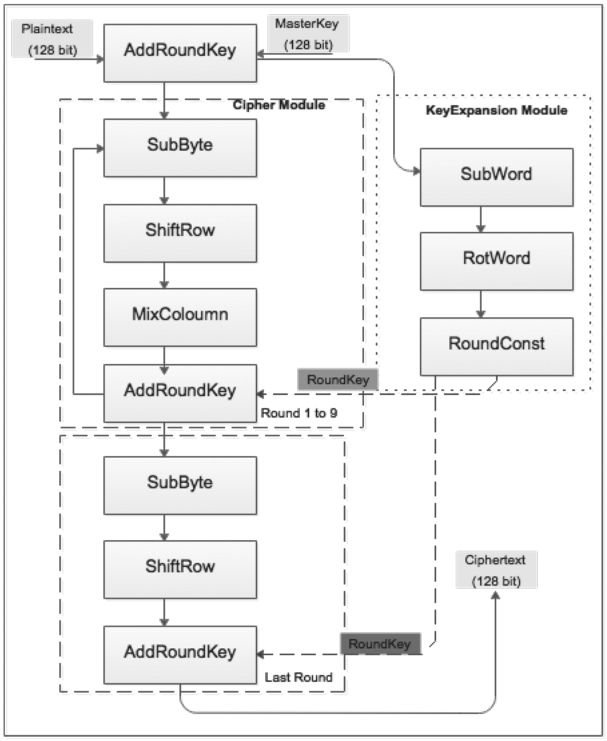


Figure 1: AES encryption algorithm structure

In general, the implementation of an AES encryption algorithm is divided into two major components: the cipher module and the key generation module.

2.2 AES Algorithm

The cipher module performs data encryption or decryption. In an AES algorithm with a 128-bit key, the cipher module does ten rounds of substitutions and permutations to encrypt the data input (plaintext). In the first nine rounds of the encryption process, the cipher module uses SubByte, ShiftRow, MixColumn, and AddRoundKey operations. In the final (tenth) round, Mixcolumn is used to complete the block encryption process. Figure 1 shows the standard structure of the AES algorithm. The Figure also shows the iterative process in combining different functions of each module with the cipher key expansion module. The input data in AES is represented as a 4 x 4 byte array and is called the state.

AddRoundKey is the initial and final function and is used to combine key information with the data under operation. The input of this function is 16 bytes of state, while 16 byte keys are obtained from the key expansion algorithm. The output value of this operation is the XOR bit between round sub key derived from master key. This function is essentially same for encryption and decryption processes. The transformation is denoted by $AddRoundKey(State, RoundKey)$.

The SubByte transformation is a non-linear byte substitution, operating on every byte of the state inde-

Table 1: Comparison of AES implementation

	ASIC	**FPGA**	**Software Based**
Parallel processing	Yes	Yes	Limited
Pipelining	Yes	Yes	Limited
Velocity	Very fast	Fast	Moderate
Resistance to tamper	Strong	Strong	Weak
Design process	Long time	Moderate	Fast
Field reprogramability	Moderate	Yes	Moderate
Lower Unit Cost	Yes	Moderate	Yes

pendently. The substitution table (or S-box) is invertible Each input of byte is independently replaced by another byte from lookup table called substitution box (S-box). There are 16 parallel S-boxes, each consisting of 8 input and 8 output values. The S-box operation is the only non-linear transformation in the AES algorithm. The transformation is denoted by $SubByte(State)$. This operation is reversible and is also used in the decryption process. The S-box design is achieved by combining two transformations. The first transformation is done by taking the inverse multiplication in the finite field $GF(2^8)$ where all the zero bit inputs are mapped to themselves. In the second part, the affine transformation is done over $GF(2)$.

$$M = \left\{\begin{matrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{matrix}\right\}, C = \left\{\begin{matrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{matrix}\right\}$$

ShiftRow function, the number of rows in the State are cyclically shifted over different offsets. In this function, the first line is not altered, but the second, third, and fourth lines are rotated by one, two, three bytes, respectively. The shifting the rows of the State over the specified offsets is denoted by:$ShiftRow(State)$. The MixColumn perform transformation each coloumn of the state matrix multiplied by a constant fixed matrix. The application of MixColoumn operation on all columns of the State is denoted by $MixColumn(State)$. This function can be written into a matrix multiplication as:

$$\begin{bmatrix} MC^{'}_{0,c} \\ MC^{'}_{1,c} \\ MC^{'}_{2,c} \\ MC^{'}_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} MC_{0,c} \\ MC_{1,c} \\ MC_{2,c} \\ MC_{3,c} \end{bmatrix}$$

2.3 Key Generation Module

The key generation module are derived from the MasterKey by means of the key schedule. This consists of two main components: the MasterKey Expansion and the SubKey Selection. The key generation algorithm is as follows:

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
Begin
word temp
i=0
while i < Nk do
    w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
    i = i+1
end while
i=Nk
while (i < Nb * (Nr+1)) do
    temp = w[i-1]
    if (i mod Nk = 0) then
        temp = SubWord(RotWord(temp))  Rcon[i/Nk]
    else if (Nk > 6 and i mod Nk = 4) then
        temp = SubWord(temp)
    end ifw[i] = w[i-Nk]  temp
    i = i + 1
end while
End
```

The KeyGeneration is a linear array of 4-byte words and is denoted by $W[Nb * (Nr + 1)]$. The first Nk words contain the MasterKey. All another words are defined recursively in terms of words with smaller indices. The key expansion function depends on the value of Nk. This function consists of three sub-modules: SubWord, RotWord, and RCon. SubWord is is a function that returns a 4-byte word in which each byte is the result of applying the AES S-box to the byte at the corresponding position in the input word. RotWord takes a cyclic permutation of those in its input such that the input $word(a, b, c, d)$ produces the output $word(b, c, d, a)$. The ratio between key length and block size in terms of words for different types of AES is shown in Table 2.

2.4 Device Optimization

Requirements of throughput, power and design need to be met by AES algorithm designers. As such, AES devices need to be optimized in order to meet these requirements. Most approaches to AES design optimization can

Table 2: Comparison of key length, block size, and AES round

AES Type	Key Length	Block Size	Round
AES 128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

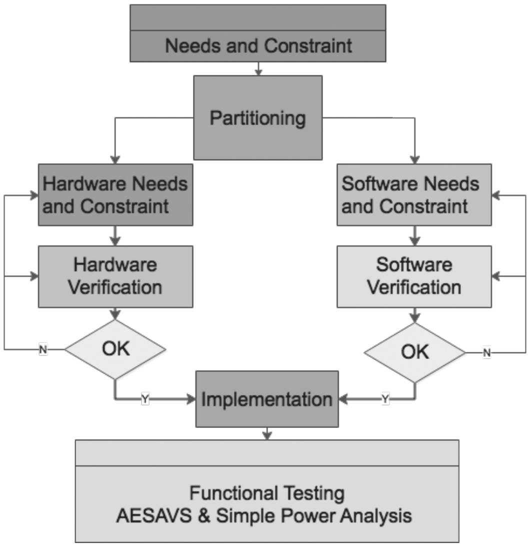


Figure 2: Design of AES device methodology

be generally divided into two: algorithm optimizations and architecture optimizations.

Algorithm optimizations use the fact that AES is based on finite fields operations. Thus, the choice that represents the finite field and composite fields is the use of isomorphisms in developing efficient and compact design [17]. Furthermore, an AES round has properties which allow the encryption and decryption processes to be done using the same method. Platforms such as the FPGA provide enables efficient implementation of some of the AES round transformations.

Meanwhile, standard architectural optimization techniques such as pipelining can also be applied to improve the hardware design throughput of AES [11]. The separation of data and key generation scheduling is helpful in improving the efficiency of an AES design [13]. This optimization technique can support all three versions of AES, namely 128.192, and 256 bits [14].

3 Methodology

This paper presents the stages involved in designing an AES encryption device. The proposed method is shown in Figure 2.

The initial stage is reviewing the aims and limitations for designing the AES cryptographic device. The aim of this research is to create an AES device implemented on an FPGA platform which will be a simulated DUT device in a DPA (differential power analysis) test. The restriction on this device is AES with 128-bit key length and an input value message with 128-bit length.

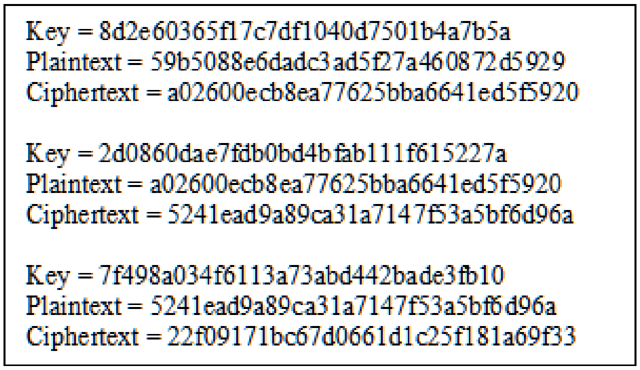
The next design stage is categorizing the identified aims and limitations into hardware and software. The initial approach is to generate a Verilog pseudocode and RTL schematic of the desired circuit in simulation software such as ModelSim. The resulting scheme can be verified using simulation software that shows the input and output waveforms of the circuit. The output is the Verilog code and a RTL schematic design for an AES cryptographic algorithm to be verified on FPGA hardware. After the aims and limitations have been verified on both hardware and software, the next step is implementation and functionality testing.

The implementation and testing process begins with designing and conducting an AES cryptographic algorithm testbench using FPGA software in Verilog. The testbench process checks whether the device gives the correct output, given a standard input. After passing the testbench, the design is ready to be implemented on an FPGA platform. The end point of the FPGA prototype is that when a Verilog model is decoded into ”gates and cables” are mapped onto a programmable logic device such as an FPGA. Tests were conducted to determine the success rate of the system that was developed. Three stages of testing were conducted:

- Functional testing of each system block.
- System-wide functional testing.
- Hardware performance measurement.

The functional testing of the encryption device was done using the Advanced Encryption Standard Algorithm Validation Suite (AESAVS). AESAVS is designed to test compliance with the NIST FIPS197 standard. This testing standard provides a security measure for an AES product. Test validation was performed to assist in detecting any errors of unintended implementation [14]. So this validation technique should not be interpreted as an evaluation or overall product security support. The AESAVS testing method has the following philosophy design:

1) AESAVS is designed to allow testing of implementation under test (IUT). AESAVS and IUT exchange data via request and response file.



```
Key = 8d2e60365f17c7df1040d7501b4a7b5a
Plaintext = 59b5088e6dadc3ad5f27a460872d5929
Ciphertext = a02600ecb8ea77625bba6641ed5f5920

Key = 2d0860dae7fdb0bd4bfab111f615227a
Plaintext = a02600ecb8ea77625bba6641ed5f5920
Ciphertext = 5241ead9a89ca31a7147f53a5bf6d96a

Key = 7f498a034f6113a73abd442bade3fb10
Plaintext = 5241ead9a89ca31a7147f53a5bf6d96a
Ciphertext = 22f09171bc67d0661d1c25f181a69f33
```

Figure 3: Example of output requirements and monte carlo test input

2) Testing is done in AESAVS using statistical sampling.

Devices are required to pass AESAVS to claim compliance to the FIPS 197 standard on AES and FIPS 140-2 standard on the security requirements for cryptographic modules.

AESAVS uses two approaches; namely, the known answer test (KAT) and Monte Carlo test (MCT). KAT tests KeySbox, Variable Key and Variable Text. MCT tests the cipher with as many as 100 pseudorandom texts. These texts are generated using an algorithm related to the mode of operation under test. An example of the three initial outputs of MCT from AES128 is shown in Figure 3.

4 AES Implementation

This study uses Verilog language as the hardware description language because of its flexibility; Verilog codes can be easily implemented in other devices without design changes. The hardware used for this research is Altera DE2 board. This board tool is used for writing, debugging, optimizing, performing simulations, and checking performance results using the simulation tools available on the Quartus IDE design software.

The results of this study are based on simulations of Altera DE2 and Quartus tools, using analysis and synthesis, RTL viewer, and state machine viewer. A design iteration method is implemented to minimize hardware utilization and adaptation performed on Altera Quartus.

Figure 4 shows a block diagram of the hardware implementation. In the block diagram is also mentioned the encryption and decryption module and key generation module.

The system clock represented by *clk* sets the crypto-processor frequency. The clock frequency dictates the operating speed of the system. In this design, the clock has two main purposes: system controller and bus or line controller. The reset function *rst* enables a return of the system to the initial condition. This function is active in low or 0 condition. At the key generation module initialization, the *start* signal states the time at which the clock starts. The *decrypt* function, one of the main functions of this system, transforms ciphertext into plaintext. This function can be changed into encryption mode (on which the function transforms plaintext into ciphertext) by changing the input value of En (encryption mode when En = 1, decryption mode when En =0). Additional input values are the 128-bit key and din which are the key and text input, respectively. These input values produce 128-bit *dout*, the same size as the input.

The key generation process is shown in Figure 6. This process generates the round keys from the master key; these round keys are only in the encryption rounds. The clk function is the input in key generation; the generated keys are stored in the internal ROM and read by encryption and decryption block for each round. The encryption/decryption module accepts a 128-bit plaintext or ciphertext input when the decrypt is off ($En = 0$). The key generation process accepts 128-bit key input, divided into 4 sections (ki_0, ki_1, ki_2, ki_3) of 32bit. In this design, more than one register is used in each round, producing ten clocks efficient for key generation.

The system design architecture is shown in Figure 7. There are at least three connected components: AES crypto-processor, personal computer, and digital sampling oscilloscope (DSO). The crypto-processor is the DUT from which side channel information would be harvested by the DSO, creating a traces curve. The PC collects the traces and performs statistical analyses to find the key by modeling the traces curve using key guesses. The DUT and the oscilloscope communicate using USB and RS232.

The results shown in Table 1 illustrate the condition of the implementation of the AES encryption algorithm on the Cyclone IV E FPGA device. Generally, the required element of this device is 8% of the total device. This shows that this system and platform can be applied to other solution problems. The small memory usage of $< 1\%$ and the required registers indicates that the AES device design meets excellent performance standards on various application forms and platforms, as well as high speed and low latency values.

The use of a small amount of area and a compact design is shown in Figure 8. This shows a slice of the usage of the FPGA EP4CE115F29C7 processor. For a small to medium sized processor, the slice usage is extremely small at 8%. This shows a potential for implementing KGS (knowledge growing system) as countermeasures for DPA/SCA.

5 DUT Testing

The AES encryption tool is built in the ECB operation mode and uses Verilog as a programming language. Description and verification was done using a ModelSim simulator - Intel FPGA Starter 10.5b edition. Quartus software performs logic synthesis, mapping, placing, and

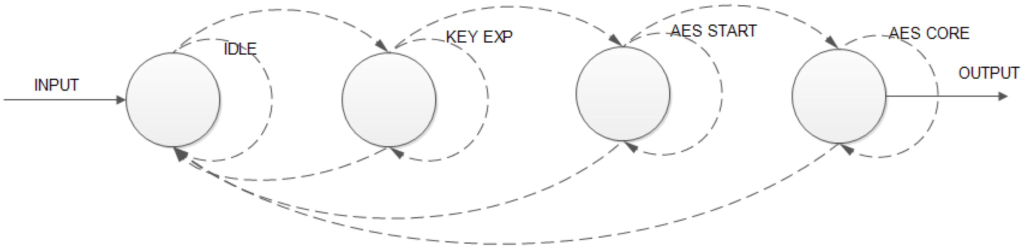
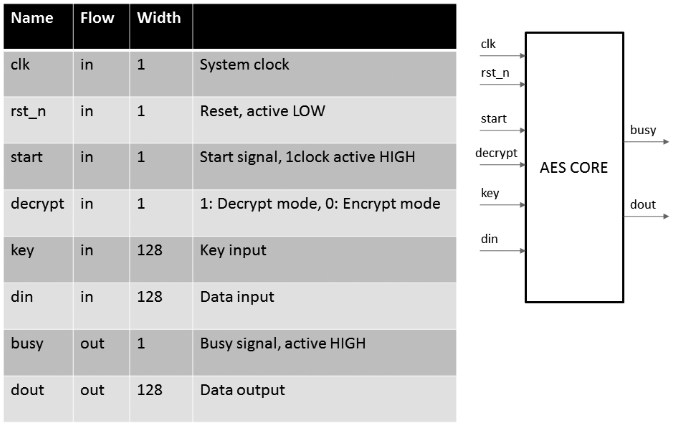


Figure 4: Finite-state of AES device

Table 3: Result comparison testing

Testing Component	Iteration Length	Encrypt/Decrypt Pass	Time
AESAVS VarKey test data for ECB AES128	128	Yes	92505 ps
AESAVS KeySbox test data for ECB	21	Yes	15465 ps
AESAVS VarTxt test data for ECB	128	Yes	92505 ps
AESAVS MCT test data for ECB	128	Yes	28044345 ps



Name	Flow	Width	
clk	in	1	System clock
rst_n	in	1	Reset, active LOW
start	in	1	Start signal, 1clock active HIGH
decrypt	in	1	1: Decrypt mode, 0: Encrypt mode
key	in	128	Key input
din	in	128	Data input
busy	out	1	Busy signal, active HIGH
dout	out	128	Data output

Figure 5: Block diagram of resulting AES device

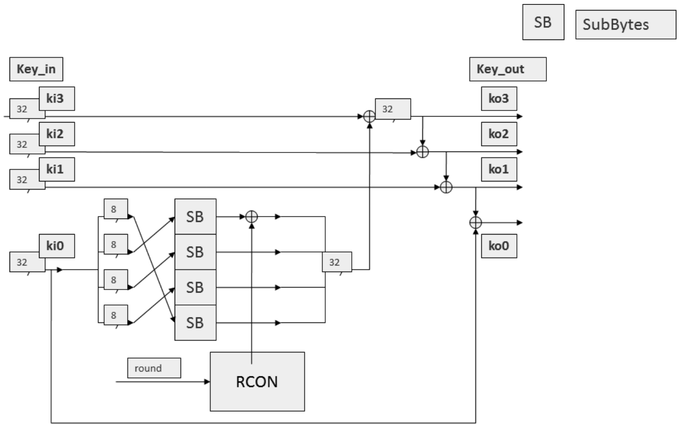


Figure 6: Key generation process

routing; Verilog pseudocodes are used for the input of this process. All built-in circuits are contained in a single FPGA device, with Quartus software determining the optimal location for a compact design. Table 4 shows the results of AESAVS testing compare with another AES design.

AESAVS test results through the KAT test (known answer test) and the MCT (Monte Carlo test) have shown results that meet the NIST FIPS197 standard and does not detect implementation errors. This validation tool has been evaluated and complies with NIST's AES FIPS197 standard.

Table 4 provides a comparison between several commonly used AES standards: Sasebo [14], SakuraX [15], and the design proposed in this paper. It shows that the proposed communication interface standard has a higher flexibility than other devices. This flexibility gives more opportunities for researchers to develop countermeasures to attacks. The proposed device has a size of 90nm in design technology, which is still larger than Sakura-X whose value is 28 nm. However, this value affects neither performance nor speed. It only affects the area/wafer of the device.

This study also performs synthetic analysis on three devices: AESFPGA, asic.ws, and our proposed design. Our proposed design has advantages over the other designs in total register, pins, and memory bits. Our proposed DUT has 277 registers, which is smaller than that of the other devices. Our proposed device uses 389 input and output pins, which is better than the AESFPGA design standard. The most important factor affecting the speed and performance of the proposed DUT is the usage of memory bits as cache. This results in a much faster speed than the other two designs.

Besides testing with KAT, we also do trace sampling analysis with SPA (simple power analysis). The purpose of this test is as a first step DPA attack techniques. Fig-

Table 4: Comparison our proposed with FPGAs as device under test (DUT)

Board	Control FPGA	Tech	Host Data Communication	Status
SASEBO [14, 15, 28]	Virtex-2 Pro	130 nm	RS232	Discontinued
SASEBO-G [14, 15, 28]	Virtex-2 Pro	130 nm	RS232,FT245RL(USB)	-
SASEBO-GII [14, 15, 28]	Spartan-3A	65 nm	FT245RL	Discontinued
SASEBO-B [14, 15, 28]	Stratix-2	90 nm	RS232,FT245RL(USB)	-
Saxura-X [14, 15]	Spartan-6	28 nm	USB	-
Our Proposed	Altera Cyclone II	90 nm	RS232,FT245RL(USB),JTAG Mode	-

Table 5: Synthesis result of AES device

Component	**Our Proposed**	AESFPGA1 [14]	ASIC.WS
Revision Name	`AES_CORE`	`SASEBO`	`ASIC`
Top-level Entity Name	`AES_CORE`	`SASEBO`	AES ASIC
Total logic elements	9,153 / 114,480 (8%)	9,023	4,940
Total registers	277	396	530
Total pins	389 / 529 (74%)	392 /529	388
Total virtual pins	0	0	0
Total memory bits	2,048 / 3,981,312 (1%)	0	0
Embedded Multiplier 9-bit elements	0 / 532 (0%)	0	0
Total PLLs	0 / 4 (0%)	0	0

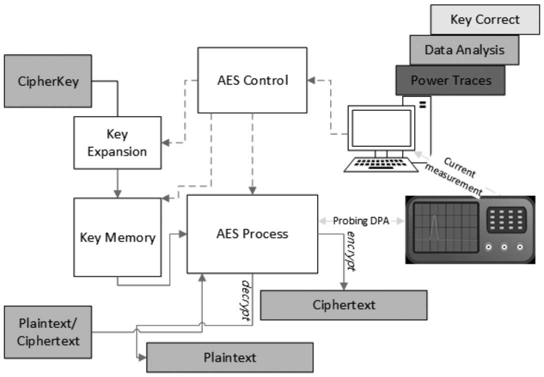


Figure 7: Architecture of proposed DUT

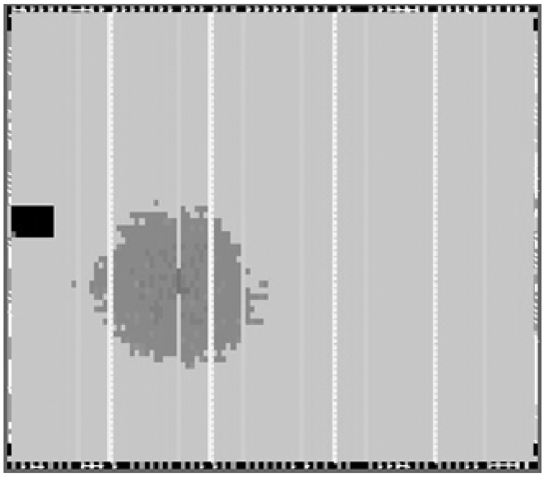


Figure 8: Use of AES slice area generated

ure 9 shows approximately 1 ms of a trace collected from a DUT performing an AES-128 encryption operation. The power consumption was sampled at 100 MHz. The trace were captured by placing a resistor in series with the devices ground line in accordance with architecture DUT Figure 7, then using an oscilloscope to measure the voltage at the ground input. The trace from a DUT shows the ten rounds clearly visible, and ready for DPA attacks.

6 Concluding Remarks

This paper presents a design for implementation of 128-bit AES FPGA. This encryption tool supports encryption and decryption processes that meet the NES AES FIPS 197 standard and the advanced encryption standard algorithm validation suite (AESAVS) testing. In future research, the AES encryption device will need to be tested against simple power analysis and differential power analysis. Currently, this DUT is still under development. It is hoped that this design would help hardware researchers in studying cryptographic attacks and countermeasures. The usage of open source and reconfigurable hardware (FPGA) allows for easy application to wider research areas, such as higher order DPA and correlation power analysis. Future development of this device is planned so that

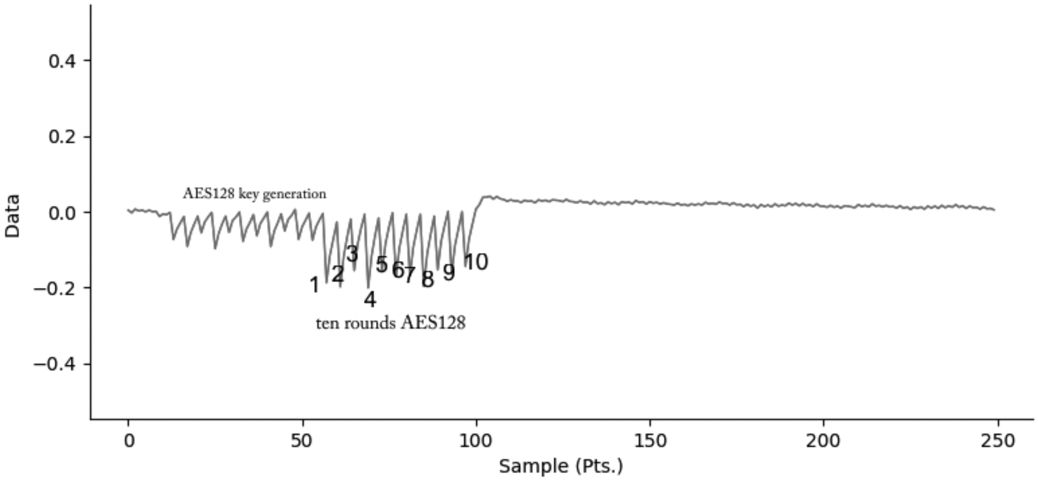


Figure 9: Power traces from a DUT performing an AES128

it could be configured to different encryption algorithms, such as RSA, ECC, DES, and BC3. Our proposed DUT also has applications in various platforms, such as smartcards, microcontrollers, and ASIC, with high throughput and latency.

Acknowledgments

The first author acknowledges support from the Indonesia Endowment Fund for Education (LPDP -Lembaga Pengelola Dana Pendidikan) scholarship, Ministry of Finance, The Republic of Indonesia. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] F. Bao, C. C. Lee, and M. S. Hwang, "Cryptanalysis and improvement on batch verifying multiple rsa digital signatures," *Applied Mathematics and Computation*, vol. 172, no. 2, pp. 1195–1200, 2006.

[2] S. S. Chawla and N. Goel, "FPGA implementation of an 8-bit AES architecture: A rolled and masked s-box approach," in *Annual IEEE India Conference (INDICON'15)*, pp. 1–6, Dec. 2015.

[3] C. Chiţu and M. Glesner, "An FPGA implementation of the AES-Rijndael in OCB/ECB modes of operation," *Microelectronics Journal*, vol. 36, no. 2, pp. 139–146, 2005.

[4] H. S. Deshpande, K. J. Karande, and A. O. Mulani, "Area optimized implementation of AES algorithm on FPGA," in *International Conference on Communications and Signal Processing (ICCSP'15)*, pp. 10–14, 2015.

[5] O. S. Dhede and S. K. Shah, "A review: Hardware implementation of AES using minimal resources on FPGA," in *International Conference on Pervasive Computing (ICPC'15)*, pp. 1–3, 2015.

[6] J. F. Dooley, *A Brief History of Cryptology and Cryptographic Algorithms*, Springer, 2013.

[7] J. D. Golić and C. Tymen, "Multiplicative masking and power analysis of AES," in *Cryptographic Hardware and Embedded Systems (CHES'02)*, pp. 198–212, 2003.

[8] J. M. Granado-Criado, M. A. Vega-Rodríguez, J. M. Sánchez-Pérez, and J. A. Gómez-Pulido, "A new methodology to implement the AES algorithm using partial and dynamic reconfiguration," *Integration, the VLSI Journal*, vol. 43, no. 1, pp. 72–80, 2010.

[9] T. Gulom, "The encryption algorithm AES-PES16-1 and AES-RFWKPES16-1 based on network PES16-1 and RFWKPES16-1," *International Journal of Electronics and Information Engineering*, vol. 3, no. 2, pp. 53–66, 2015.

[10] T. Gulom, "The encryption algorithm GOST28147-89-PES16-2 and GOST28147-89-RFWKPES16-2," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 1–11, 2017.

[11] M. Hutter, M. Kirschbaum, T. Plos, J. M. Schmidt, and S. Mangard, "Exploiting the difference of side-channel leakages," in *Constructive Side-Channel Analysis and Secure Design*, pp. 1–16, 2012.

[12] M. S. Hwang, C. T. Li, J. J. Shen, Y. P. Chu, "Challenges in e-government and security of information", *Information & Security: An International Journal*, vol. 15, no. 1, pp. 9–20, 2004.

[13] J. N. Jr, "Analysis of venkaiah *et al.*'s AES design," *International Journal of Network Security*, vol. 9, no. 3, pp. 285–289, 2009.

[14] T. Katashita, Y. Hori, H. Sakane, and A. Satoh, "Side-channel attack standard evaluation board sasebo-w for smartcard testing," *Power*, vol. 3, pp. 400, 2012.

[15] T. Katashita, A. Satoh, K. Kikuchi, H. Nakagawa, and M. Aoyagi, "Evaluation of dpa characteristics of sasebo for board level simulations," in *International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE'10)*, vol. 36, pp. 39, 2010.

[16] G. Khedkar, D. Kudithipudi, and G. S. Rose, "Power profile obfuscation using nanoscale memristive devices to counter DPA attacks," *IEEE Transactions on Nanotechnology*, vol. 14, pp. 26–35, Jan. 2015.

[17] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Moderator-Ravi, "Security as a new dimension in embedded system design," in *Proceedings of the 41st annual Design Automation Conference*, pp. 753–760, 2004.

[18] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems," in *Advances in Cryptology (CRYPTO'96)*, pp. 104–113, 1996.

[19] P. C. Kocher, J. M. Jaffe, and B. C. Jun, *Differential Power Analysis*, US Patent 7, 599, 488, Oct. 6, 2009..

[20] I. C. Lin, M. S. Hwang, C. C. Chang, "A new key assignment scheme for enforcing complicated access control policies in hierarchy", *Future Generation Computer Systems*, vol. 19, no. 4, pp. 457–462, May 2003.

[21] M. Masoumi, P. Habibi, and M. Jadidi, "Efficient implementation of masked AES on side-channel attack standard evaluation board," in *International Conference on Information Society (i-Society'15)*, pp. 151–156, 2015.

[22] A. Mersaid, T. Gulom, "The encryption algorithm AES-RFWKIDEA32-1 based on network RFWKIDEA32-1," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 1–11, 2016.

[23] S. D. Putra, A. S. Ahmad, and S. Sutikno, "Dpa-countermeasure with knowledge growing system," in *International Symposium on Electronics and Smart Devices (ISESD'16)*, pp. 16–20, 2016.

[24] S. D. Putra, A. S. Ahmad, and S. Sutikno, "Power analysis attack on implementation of DES," in *International Conference on Information Technology Systems and Innovation (ICITSI'16)*, pp. 1–6, 2016.

[25] S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper resistance mechanisms for secure embedded systems," in *17th International Conference on VLSI Design*, pp. 605–611, 2004.

[26] Y. Souissi, S. Guilley, S. Bhasin, and J. L. Danger, "Common framework to evaluate modern embedded systems against side-channel attacks," in *IEEE International Conference on Technologies for Homeland Security (HST'11)*, pp. 86–91, 2011.

[27] G. Tuychiev, "The encryption algorithms GOST28147-89-IDEA8-4 and GOST28147-89-RFWKIDEA8-4," *International Journal of Electronics and Information Engineering*, vol. 6, no. 2, pp. 59-71, 2017.

[28] R. Velegalati and J. P. Kaps, "Introducing FOBOS: Flexible open-source bOard for side-channel analysis," in *Third International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE'12)*, Work in Progress Session, 2012.

[29] J. Wu, Y. Shi, and M. Choi, "Measurement and evaluation of power analysis attacks on asynchronous S-box," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, pp. 2765–2775, Oct. 2012.

[30] W. Yu and S. Köse, "A voltage regulator-assisted lightweight AES implementation against dpa attacks," *IEEE Transactions on Circuits and Systems*, vol. 63, pp. 1152–1163, Aug. 2016.

Biography

Septafiansyah Dwi Putra received the B.S. and M.S. degrees in electrical engineering and informatics from Lampung University and Institut Teknologi Bandung, in 2011 and 2014, respectively. Currently, he is working toward the Ph.D. degree in CAIRG-Research Group, School of Electrical Engginering and Informatics, Institut Teknologi Bandung, Indonesia. His current research interests are computer security, cognitive artificial intelligence and hardware security. He is currently associated with Management of Informatics - Politeknik Negeri Lampung as lecturer.

Ma'muri received the B.S. degree in electrical engineering from Institut Teknologi Bandung (ITB), Bandung, Indonesia, in 2004, respectively. He is currently the Master Student of School of Electrical Engineering and Informatics ITB. His research interests focus on the design and analysing hardware security.

Sarwono Sutikno received B.S in Electronics degree from Institute Technoloy of Bandung, Bandung, Indonesia, in 1984, and received the Master of Engineering degree and Doctor of Engineering degree in Integrated System from Tokyo Institute of Technology, Tokyo, Japan in 1990 and 1994, respectively. His research interests focus on implementation of cryptographics algorithms in Integrated Circuits including Embedded System Security. His Security Engineering focus includes Information Security Management System. He holds several professional certifications including Certified Information System Auditor and ISMS Provisional Auditor, he is also appointed ISACA Academic Advocate. Currently, He is advisor to the Corruption Eradication Commission Republic of Indonesia.

Yusuf Kurniawan received the B.S. degree,master degree, and doctoral degree in electrical engineering from Institut Teknologi Bandung (ITB), Bandung, Indonesia, in 1994, 1997, 2007, respectively. He is currently as lecture of School of Electrical Engineering and Informatics ITB. His research interests focus on the design of block cipher and cryptology.

Adang Suwandi Ahmad (ASA) received his engineering degree (Ir.) in Electrical Engineering from ITB in 1976, Diplome Etude Approfondi Signaux et Bruits (DEA) option Electronique, and Docteur Ingenieur Signaux et Bruits option Electronique (Dr.-ing) from Universite des Sciences du Languedoc Montpellier, France in 1978 and July 1980 respectively. He became Institut Teknologi Bandungs Professor in Intelligent Electronics Instrumentation System in 2000. ASA past researches were in Electronics Instrumentation Systems (Devices and Systems) and Intelligent Electronics Systems/Artificial Intelligence. Cooperation with Navy Research Service (1992) had yielded a War Game Simulator. He founded Intelligent System Research Group(ISRG)ITB in1993. Now he focuses in information sciences, intelligent computations, and intelligent-based systems. He is the former Dean of the School of Electrical Engineering and Informatics ITB, Bandung, Indonesia.

Certificate-based Smooth Projective Hashing and Its Applications

Sujuan Li[1], Yi Mu[2], and Mingwu Zhang[3]
(Corresponding author: Sujuan Li)

School of Mathematical and Physical Sciences, Nanjing Tech University[1]
No.30 Puzhu Road Pukou District, Nanjing, China
(Email: lisujuan1978@126.com)
School of Computer and Information Technology, University of Wollongong Australia[2]
Northfields Ave, Wollongong, Australia
School of Computer Science, Hubei University of Technology[3]
No.28 Nanli Road Hongshan District, Wuhan, China

(Received Oct. 21, 2016; revised and accepted Feb. 1, 2017)

Abstract

Smooth projective hashing was firstly introduced by Cramer and Shoup (EuroCrypt'02) as a tool to construct efficient chosen-ciphertext-secure public key encryption schemes. Since then, they have found many other applications, such as password-based authenticated key exchange, oblivious transfer, zero-knowledge arguments *et al.* Certificate-based encryption (CBE) not only eliminates third-party queries and heavy certificate management problem in traditional public-key encryption, but also solves key escrow problem for identity-based encryption. We introduce the new concept of certificate-based smooth projective hashing (CB-SPH). Under the security model for the leakage-resilient certificate-based encryption (LR-CBE), we show how to construct a general leakage-resilient certificate-based encryption scheme using the certificate-based smooth projective hashing. Based on these theoretical constructions, we present two concrete CB-SPH instantiations under the DBDH assumption and the DLWE assumption respectively. Based on these CB-SPH instantiations, we can construct leakage resilient CBE schemes.

Keywords: Certificate-based Smooth Projective Hashing; DBDH; DLWE; Key-leakage Resilient

1 Introduction

Traditional cryptographic schemes assume that the secret keys are completely hidden from the adversaries. However side-channel attacks [21, 36] and cold boot attacks [3] indicate that the conventional attack model fails to capture some attacks in the real world. We classify these attacks as key leakage attacks in which the attackers may obtain some partial information about the secret states of the cryptosystems. To stand against such attacks, there has been a surge of interest in creating leakage resilient cryptographic schemes [3, 5, 8, 10, 13, 15, 22, 23, 26, 28, 32, 33, 35, 37]. The feature of a leakage-resilient cryptosystem is that it remains secure even when some secret internal information including the secret key is leaked to the adversary.

Smooth projective hashing (SPH) was firstly introduced by Cramer and Shoup [12]. Originally it is a tool for constructing adaptively chosen ciphertext (CCA2) secure public key encryption (PKE). Lately SPH was found that it can be applied to construct different cryptographic schemes such as password-based authenticated key exchange, oblivious transfer, zero-knowledge arguments *et al.* In Crypto'09 Naor and Segev [32] found that SPH can be applied to the leakage resilient PKE. Under the subset membership problem (SMP), they presented a general construction of leakage-resilient PKE scheme with the help of universal SPH creatively. They extended the framework of key-leakage to the setting of chosen ciphertext attacks (Akavia *et al.* [3] formalized the first framework for modeling the security of leakage-resilient PKE). Based on the Decisional Deffie-Hellman (DDH) assumption they gave two practical PKE schemes against key leakage with 1/4 and 1/6 leakage ratio respectively. Based on Naor and Segev's work [32], there are more researches paid on the leakage-resilient PKE afterwards. Nguyen *et al.* [33] explored stateless/stateful leakage-resilient public key encryption from the SPH. Kurosawa *et al.* [23] presented another general method of constructing CCA-secure PKE with key leakage, which is based on the universal$_2$ SPH. They also gave two concrete public key encryption schemes under the DCR assumption and the DLIN assumption respectively. In AsiaCrypt'13 Qin and Liu [35] presented a new general construction of PKE scheme against leakage-resilient chosen-ciphertext attacks (LR-CCA), from any SPH and any one-time lossy filter (OT-LF).

In constructing Identity-based encryption (IBE) schemes against key leakage attacks, Alwen *et al.* [5] firstly gave the concept of ID-based smooth projective hashing (IB-SPH) and presented three IB-SPH instantiations derived from existing IBE schemes [7, 18, 19]. Based on these instantiations they also got three leakage resilient IBE schemes. With the help of the dual system encryption technology, Lewko *et al.* [24] got fully secure IBE, HIBE, and ABE systems which are resilient to bounded leakage from each of many secret keys per user, as well as many master keys. Li *et al.* [27] gave a leakage resilient IBE scheme based on the IB-SPH extracted from Coron's IBE scheme [11].

1.1 Our Motivation

Smooth projective hashing plays an important role in public key cryptosystems. Informally, a smooth projective hashing is a family of keyed hash functions. Its special construction achieves many applications such as constructing chosen-ciphertext-secure encryption, leakage-resilient encryption *et al.* From the aspect of assemble construction, many research works have been paid to the properties of the language such as conjunction or disjunction of languages [1, 6] and the properties of the smooth projective hash such as homomorphic smooth projective hashing [41] or dual projective hashing [40].

The certificate-based encryption (CBE) was firstly proposed by Gentry [17] in EuroCrypt'03. In CBE schemes, a user produces public and private keys, and applies a corresponding certificate which is given from the trusted certificate authority (CA). It is possible to obtain a plaintext only in the case that the user has the private key and certificate simultaneously. CBE not only eliminates third-party queries and heavy certificate management problem in traditional public-key encryption, but also solves key escrow problem for identity-based encryption. CBE has attracted more concern due to its advantages. Recently, many CBE schemes [16, 29, 31, 42] have been proposed. In the key leakage resilient setting, Yu *et al.* [39] proposed the first leakage resilient certificate-based public key encryption using the dual system encryption technology. But their construction is in the composite order group which costs much more than in a prime order group. Afterwards Yu *et al.* [38] proposed another leakage resilient certificate-based public key encryption under the DBDH assumption in the random oracle. Li *et al.* [25] proposed a continuous leakage resilient certificate-based encryption with the help of secret sharing technology.

Focusing on the construction of the secret key in the smooth projective hashing, we can see some useful new smooth projective hashing which can be operated under some new conditions. Yang *et al.* [37] proposed an updatable smooth projective hashing which can be used to design general constructions of public key encryption schemes against continuous key leakage. Considering the secret key is delegated by the key generate center, Alwen *et al.* [5] introduce an identity-based smooth projective hashing which can be used to design the leakgae resilient identity-based encryption schemes.

To explore the smooth projective hashing with applications in certificate-based cryptosystems and obtain the leakage resilient certificate-based public key encryption schemes in the prime order group, we propose the new notion of certificate-based smooth projective hashing, borrowing the ideas of Yang et al [37] and Alwen et al [5]. In this paper we focus on the certificate-based smooth projective hashing and its application in leakage resilient encryptions which can also be extended to the continuous leakage resilient settings.

1.2 Our Contribution

In brief, our contribution is described as follows:

1) In the certificate-based settings we firstly give a definition of generalized certificate-based smooth projective hashing (CB-SPH). In order to guarantee its security we verify the smooth and projective properties.

2) Based on the definition and security properties of CB-SPH, we show how to convert smooth CB-SPH to leakage resilient one and show how to construct leakage-resilient certificate-based encryption schemes.

3) As a concrete example, in a prime-order group we present the first practical CB-SPH construction under the DBDH assumption in the standard model. To achieve a leakage-resilient certificate-based encryption, the construction using CB-SPH tool is much more efficient and practical comparing with the construction using dual system encryption technology in the composite order group.

4) Under the decisional learning with errors assumption, we firstly present a lattice-based CB-SPH instantiation which also can be transferred into a leakage resilient certificate-based encryption.

1.3 Organization

The rest of the article is organized as follows. We review some preliminaries that are used in this article in Section 2. In Section 3, we present the general construction of the certificate-based smooth projective hashing with projection and smoothness properties. In Section 4, we introduce the security model and the generic construction for the leakage-resilient certificate-based encryption. Two concrete certificate-based smooth projective hashings based on the DBDH assumption and the DLWE assumption respectively are shown in Section 5. Lastly, we give a conclusion and future work in Section 6.

2 Preliminaries

In this section, we present some basic notions and tools that will be used in our constructions and security proofs. We formally state some decisional assumptions and present the notion of average-case strong randomness extractors.

2.1 Computational Assumptions

Let $\mathsf{BLGroupGen}$ be a PPT algorithm that takes as input a security parameter κ and output a tuple (G, G_1, g, e). Let G and G_1 be the two cyclic groups of order p for some large prime p. A map $e : G\times G \rightarrow G_1$ is a bilinear pairing. Let g be a random generator of G. The following DBDH assumption is given in $(G, G_1, g, e) \leftarrow \mathsf{BLGroupGen}(\kappa)$.

Definition 1. *(Decisional Bilinear Diffie-Hellman Assumption) We define the decisional bilinear Diffie-Hellman (DBDH) problem as: Given (G, g, g^a, g^b, g^c) and a random element $Z \in G_1$ as input, output 1 if $Z = e(g, g)^{abc}$ and output 0 otherwise. We say that the (t, ϵ)-DBDH assumption holds if no t-time algorithm has advantage at least ϵ in solving the DBDH problem.*

Definition 2. *(Decisional Learning With Errors Assumption) We define the decisional learning with errors (DLWE) problem as: For an integer $p \geq 2$ and some probability distribution χ over Z_p, an integer dimension $n \in Z^+$, and a vector $\boldsymbol{s} \in Z_p^n$, define $A_{s,\chi}$ as the distribution over $Z_p^n \times Z_p$ of the variable $(\boldsymbol{a}, \boldsymbol{a}^T\boldsymbol{s} + t)$ where $\boldsymbol{a} \in_R Z_p^n$ and $t \leftarrow \chi$. The decisional learning with errors (DLWE) assumption holds if the distribution $A_{s,\chi}$ and the uniform distribution over $Z_p^n \times Z_p$ are computationally indistinguishable, where $\boldsymbol{s} \in_R Z_p$.*

2.2 Random Extractor

The statistical distance between two random variables X and Y over a finite domain Ω is $\mathrm{SD}(X, Y) = \frac{1}{2}\sum_{\omega\in\Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$. We write $X \approx_\epsilon Y$ to denote $\mathrm{SD}(X, Y) \leq \epsilon$, and $X \approx Y$ to denote that the statistical distance is negligible. The min-entropy of a random variable X is $H_\infty(X) = -\log(max_x \Pr[X = x])$.

We use the notion of average min-entropy [14] which captures the remaining unpredictability of a random variable X conditioned on another random variable Y, formally defined as:

$$\widetilde{H}_\infty(X|Y) = -\log(E_{y\in Y}[2^{-H_\infty(X|Y=y)}])$$

where $E_{y\in Y}$ denotes the expected value over all values of Y .

Lemma 1. *[14] For any random variables X, Y, Z, if Y has 2^r possible values, then*

$$\widetilde{H}_\infty(X|(Y, Z)) \geq \widetilde{H}_\infty(X|Z) - r.$$

Specially,

$$\widetilde{H}_\infty(X|Y) \geq H_\infty(X) - r.$$

Definition 3. *[14] A function Ext:$\{0, 1\}^u \times \{0, 1\}^t \rightarrow \{0, 1\}^v$ is an average-case (m, ϵ)-strong extractor if for all pairs of random variables (X, Z) such that $X \in \{0, 1\}^u$ and $\widetilde{H}_\infty(X|Z) \geq m$ it holds that*

$$SD((\mathsf{Ext}(X, R), R, Z), (U_v, R, Z)) \leq \epsilon.$$

where R is uniform in $\{0, 1\}^t$.

The definition of universal hashing [5] and the leftover-hash lemma [34] are given as follows.

Definition 4. *(ρ-Universal Hashing) A family $\mathcal{H}$, consisting of deterministic functions $h : \{0, 1\}^u \rightarrow \{0, 1\}^v$, is ρ-universal hash family if for any $m_1 \neq m_2 \in \{0, 1\}^u$ we have $Pr_{h\leftarrow\mathcal{H}}[h(m_1) = h(m_2)] \leq \rho$.*

Lemma 2. *(Leftover-Hash Lemma [34]) Assume that the family $\mathcal{H}$ of functions $h : \{0, 1\}^u \rightarrow \{0, 1\}^v$ is a ρ-universal hash family. Then the randomized extractor $\mathsf{Ext}(x, r) = h(x)$, where h is uniform over $\mathcal{H}$, is an (m, ϵ)-extractor as long as $m \geq v+2\log(1/\epsilon)$ and $\rho \leq \frac{1}{2^v}(1+\epsilon^2)$.*

This lemma implies that the universal hash functions are also good extractors.

3 Certificate-based Smooth Projective Hashing (CB-SPH)

As we have discussed in the introduction, smooth projective hashing serves as a good framework to unify many PKE schemes based on decisional assumptions. Before we introduce the new notion of CB-SPH, we firstly recall the basic conception about SPH.

3.1 Smooth Projective Hashing

The Smooth projective hashing (SPH) consists of two ingredients, namely the subset membership problem (SMP), which will be extended to the distribution distinguishable problem (DDP) for including the lattice-based hardness assumption, and the projective hash function (PHF).

3.1.1 Subset Membership Problem

The SMP defines a set X and a language $L \subset X$, from which a member x can be efficiently sampled with a witness w. We give the formal definition of SMP in [12].

Definition 5. *(Subset Membership Problem). A subset membership problem S specifies a collection $(S_\kappa)_{\kappa\geq0}$ of distributions. For every value of a security parameter $\kappa \geq 0$, S_κ is a probability distribution over instance descriptions.*

An instance description $\Lambda = (X, W, PK, L, R)$ specifies the following:

- Finite non-empty sets X, W, PK, and two collections of distributions $L = (L_{pk})_{pk \in PK}$ and $X \setminus L = (X \setminus L_{pk})_{pk \in PK}$ over X.
- A collection of binary relations $R = (R_{pk})_{pk \in PK}$ defined over $X \times W$. For $x \in X$ and $w \in W$ and some $pk \in PK$ such that $x, w \in R_{pk}$, we say that w is a witness for x.

A subset membership problem is hard if it is computationally impossible to distinguish random members $x \in L$ from random non-members $x \in X \setminus L$.

3.1.2 Distribution Distinguishable Problem

To *include lattice-based PKE scheme* which implement the smooth projective hashing technique, we borrow the definition of distribution distinguishable problem (DDP) in [9] which relaxes some restrictions (More details refer to [9]).

Definition 6. *(Distribution Distinguishable Problem). A distribution distinguishable problem* D *specifies a collection* $(D_\kappa)_{\kappa \geq 0}$ *of distributions. For every value of a security parameter* $\kappa \geq 0$, D_κ *is a probability distribution over instance descriptions.*

An instance description $\Gamma = (X, W, PK, A, B, R)$ specifies the following:

- Finite non-empty sets X, W, PK, and two collections of distributions $A = (A_{pk})_{pk \in PK}$ and $B = (B_{pk})_{pk \in PK}$ over X where $X = A \cup B$.
- A collection of binary relations $R = (R_{pk})_{pk \in PK}$ defined over $X \times W$. For $x \in X$ and $w \in W$ and some $pk \in PK$ such that $x, w \in R_{pk}$, we say that w is a witness for x.

$\Gamma = (X, W, PK, A, B, R)$ indicates that the instance Γ specifies X, W, PK, A, B and R. D provides the three following algorithms:

SampDDP(κ): Input a security parameter κ, and output the public and secret key pair (pk, sk) and an instance description Γ according to the distribution D_κ.

SampA(pk): Output $x \leftarrow A_{pk}$ along with a witness $w \in W$ such that $(x, w) \in R_{pk}$. This is sampling with witness algorithm.

SampB(pk): Output $x \leftarrow B_{pk}$. This is sampling without the witness algorithm.

It is only requiring that algorithms SampDDP and SampA should be efficient. A distribution problem D is said to be hard if A_{pk} and B_{pk} are computationally indistinguishable for any probability polynomial-time adversary.

3.1.3 Projective Hash Function

The PHF with projection $\alpha : SK \rightarrow PK$ is a family of hash functions H indexed by SK with domain X in SMP or domain A in DDP. For we will give a lattice-based SPH in the rest of this paper, we mainly discuss in the range of DDP.

Definition 7. *(Projective Hash Function). Let* X, Y, SK, PK *be finite non-empty sets, and* A_{pk} *be a collection of distributions indexed by* PK*. Here* X, PK, A_{pk} *are defined as in DDP above. Let* $H = \{H_{sk} : X \rightarrow Y\}_{sk \in SK}$ *be a family of functions indexed by* SK*. Let* $\alpha : SK \rightarrow PK$ *be a projection from* SK *to* PK*. We say* $\mathsf{H} = (H, SK, PK, X, A_{pk}, Y, \alpha)$ *a projective hash function if for any* $sk \in SK$ *and* $pk = \alpha(sk)$*, the action of* H_{sk} *on* $x \leftarrow A_{pk}$ *is approximately determined by* $\alpha(sk)$.

3.1.4 Generalized Smooth Projective Hashing

A generalized smooth projective hashing which encompasses the lattice-based smooth projective hashing technology combines DDP D with PHF H as following four algorithms:

Setup(κ): Run SampDDP(κ) to generate a master public/secret key pair (mpk, msk) and an instance description $\Gamma = (X, W, PK, A, B, R)$ of D, pich a projective hash function $\mathsf{H} = (H, SK, PK, X, A_{pk}, Y, \alpha)$. mpk will be used in the following algorithms.

KeyGen(κ): Pick $sk \leftarrow_R SK$, compute $pk \leftarrow \alpha(sk)$. Output the public/secret key pair (pk, sk).

Priv(sk, x): Take as input a private key sk and $x \in X$, and output $y \in Y$ such that $y = H_{sk}(x)$. It is the private evaluation algorithm.

Pub(pk, x, w): Take as input pk and $x \in A_{pk}$ with a witness $w \in W$, and output $y \in Y$. It is the public evaluation algorithm.

In the rest of this paper we still call the generalized smooth projective hashing smooth projective hashing (SPH) for short.

3.2 Certificate-based Smooth Projective Hashing

In this part we will introduce the formal definition of certificate-based smooth projective hashing (CB-SPH) and define some properties for CB-SPH. Firstly we describe the DDP and PHF in the certificate-based settings.

3.2.1 Distribution Distinguishable Problem

We only extend the algorithm SampDDP as below.

SampDDP(κ): Input a security parameter κ, and output a master public and secret key pair (mpk, msk) and an instance description Γ according to the distribution D_κ.

Therefore, the instance description Γ needs to be attached by the master public key set MPK. We get $\Gamma = (X, W, MPK, PK, A, B, R)$.

3.2.2 Projective Hash Function

The definition of projective hash function are the same as that of SPH besides H is added the master public key set MPK and the user's identity set ID i.e. $\mathsf{H} = (H, SK, MPK, ID, PK, X, A_{pk}, Y, \alpha)$.

3.2.3 Certificate-based Smooth Projective Hashing

A certificate-based smooth projective hashing (CB-SPH) P combines DDP D with PHF H in certificate-based settings. It also includes the five algorithms as follow:

Setup(κ): The authenticated center (CA) runs $\mathsf{SampDDP}(\kappa)$ to generate a master public/secret key pair (mpk, msk). The following algorithms all take mpk as input.

UserKeyGen(id): The user takes as input the master public key mpk and the identity id. It outputs the user's private key $sk_{id} = \sigma_1(id)$. Then using the master public key mpk, the identity id and the user private key sk_{id}. It outputs the user's public key $pk_{id} = \alpha(sk_{id})$.

CerGen(msk, id): For an identity id, the CA first calculates $H(id, pk_{id}) = id'$. Then, it takes as input the master public key mpk, the master secret key msk, id' and the public pk_{id}. It outputs the user?s certificate $Cert_{id} = \sigma_2(id, msk)$.

Pub(pk, id, x, w): It takes as input $id \in ID$, $x \in A_{pk}$ and a witness $w \in W$ for x, and output $y \in Y$. It is the public evaluation algorithm.

Priv($x, sk_{id}, Cert_{id},$): It takes as input user's private key sk_{id}, user's certificate $Cert_{id}$ and $x \in X$, and output $y \in Y$ such that $y = H_{sk_{id}, Cert_{id}}(x)$. It is the private evaluation algorithm.

In the CB-SPH structure, we extend the KeyGen algorithm according to the certificate-based requirements. The user's decryption key is divided into two parts. One is the user's private key generated by the user himself. And another one is the user's certificate produced by the CA. Under such extension, the private key also need to keep the projective connection with the public key.

We require a certificate-based smooth projective hashing to satisfy the following properties.

1) **Soundness.**
For any $id \in ID$, the user's private key sk_{id}, the user's certificate $Cert_{id}$, the (master) public key mpk, pk and $x \leftarrow A_{pk}$, we have

$$\text{Priv}(x, sk_{id}, Cert_{id}) = \text{Pub}(id, pk, x, w).$$

2) **Indistinguishability**.
We define the following indistinguishable game which is called IND game for short. The interactive game between the adversary $\mathcal{A}$ and the challenger $\mathcal{B}$ is described as follows.

Setup: The challenger $\mathcal{B}$ runs this algorithm to generate the master public key mpk and master secret key msk respectively. $\mathcal{B}$ sends mpk and *even* msk to $\mathcal{A}$.
Phase 1: $\mathcal{A}$ maintains two lists: L_{key}, L_{Cert} and performs the following queries in an adaptive fashion in this phase.

- $\mathsf{Private\ Key\ queries}$: $\mathcal{A}$ produces an identity id and requests the corresponding private key sk_{id}. If the item for identity id does not exist in the list L_{Key}, $\mathcal{B}$ runs the $\mathsf{UserKeyGen}$ algorithm to generate the user's private key $sk_{id} = \sigma_1(mpk, id)$.
- $\mathsf{Certificate\ queries}$: The adversary $\mathcal{A}$ asks the certificate $Cert_{id}$ for the identity id. If the item for identity id does not exist in the list L_{Cert}, B runs the UserKeyGen algorithm to generate where $Cert_{id} = \sigma_2(mpk, id, msk)$

Challenge Stage: The adversary $\mathcal{A}$ selects an arbitrary challenge identity $id^* \in ID$ randomly and possibly one for which it has seen the private key sk_{id^*} and the certificate $Cert_{id^*}$. The challenger $\mathcal{B}$ chooses $\beta \leftarrow \{0, 1\}$ randomly.
If $\beta = 1$, the challenger computers $(x, w) \leftarrow \mathsf{SampA}(mpk, id^*, pk_{id^*})$.
If $\beta = 0$, the challenger computers $x \leftarrow \mathsf{SampB}(mpk, id^*, pk_{id^*})$.
The challenger gives x to the adversary $\mathcal{A}$.
Phase 2: $\mathcal{A}$ makes a sequence of queries with $id \in ID$ adaptively as in phase 1.
Output: The adversary $\mathcal{A}$ output a bit $\beta' \in \{0, 1\}$. We say that $\mathcal{A}$ wins the game if $\beta' = \beta$.

Note that, during the **setup** phase, the challenger $\mathcal{B}$ sends the master key msk to the attacker $\mathcal{A}$ because $\mathcal{A}$ can even know the private key and the user's certificate in the following stage. We define the advantage of $\mathcal{A}$ in distinguishing honest/dishonest ciphertexts to be

$$\text{Adv}^{IND}_{CB-SPH,\mathcal{A}}(\kappa) = |\Pr[\mathcal{A} \text{ wins}] - \tfrac{1}{2}|.$$

Definition 8. *(Indistinguishability) A CB-SPH satisfies the indistinguishability property if no polynomially-time adversary $\mathcal{A}$ has a non-negligible advantage in the above game.*

3) **Projection**
A CB-SPH is projective if for any $id \in ID$,

$$\Pr[\text{Pub}(id, pk, x, w) \neq \text{Priv}(x, sk_{id}, Cert_{id})] \leq \text{negl}(\kappa)$$

where w is a witness for x and $sk_{id}, Cert_{id}$ is the user private key and certificate respectively. The probability is taken over the choice of $x \leftarrow A_{pk}$.

4) **Smoothness/Leakage-Smoothness**.
The following two properties are mainly ensuring that there are many possibilities for $\mathsf{Pub}(x, \cdot)$ of an dishonest ciphertext $x \leftarrow B_{pk}$, which are left undetermined by the public parameters of the system.

Definition 9. *(ϵ-Smooth CB-SPH) We say that a CB-SPH is ϵ-smooth if for any $id \in ID$*

$$\boldsymbol{SD}((R, id, x, y), (R, id, x, y')) \leq \epsilon,$$

where R is the ensemble of (mpk, msk), $x \leftarrow B_{pk}, y \leftarrow$Priv$(x, sk_{id}, Cert_{id})$ and $y' \leftarrow U_Y$.

Definition 10. *(l-Leakage-Resilient ϵ-Smooth CB-SPH) We say that a CB-SPH is l-leakage-resilient ϵ-smooth if for any fixed parameters produced by the above algorithms of CB-SPH and any possibly randomized function $f(\cdot)$ with l-bit output, we have:*

$$\boldsymbol{SD}((R, id, f(sk_{id}, Cert_{id}), x, y),$$
$$(R, id, c, f(sk_{id}, Cert_{id}), x, y')) \leq \epsilon$$

where R is the ensemble of (mpk, msk), $x \leftarrow B_{pk}, y \leftarrow$$\mathsf{Priv}$$(x, sk_{id}, Cert_{id})$ and $y' \leftarrow U_Y$.

3.3 Generic Construction of Leakage-Resilient CB-SPH

We now show how to convert a CB-SPH (Setup, $\mathsf{UserKeyGen}$, CerGen, Priv, Pub) into a leakage-resilient one using an average-case randomness extractor Ext: $Y \times S \rightarrow \{0,1\}^v$ with seeds set $S = \{0,1\}^\mu$. We modified the algorithms SampA and SampB of $\mathsf{DDP\ D}$ as follows:

- $\overline{\mathsf{SampA}}(pk)$: sample $(x, w) \leftarrow$SampA(pk), pick a seed $d \leftarrow_R \{0,1\}^\mu$, and output $\bar{x} = (x, w, d)$.
- $\overline{\mathsf{SampB}}(pk)$: sample $x \leftarrow$ SampB(pk), pick a seed $d \leftarrow_R \{0,1\}^\mu$, and output $\bar{x} = (x, d)$.

We keep the algorithms Setup, $\mathsf{UserKeyGen}$ and CerGen unchanged, define:

- $\overline{\mathsf{Priv}}(\bar{x}, sk_{id}, Cert_{id})$: parse $\bar{x}$ as (x, w, d), compute $y \leftarrow \mathsf{Priv}(x, sk_{id}, Cert_{id})$, and output $\bar{y} \leftarrow\mathsf{Ext}(y, d)$.
- $\overline{\mathsf{Pub}}(id, pk, \bar{x}, w)$: parse $\bar{x}$ as (x, w, d), compute $y \leftarrow \mathsf{Pub}(id, pk, x, w)$, and output $\bar{y} \leftarrow\mathsf{Ext}(y, d)$.

We will show a theorem which instruct that the transformed CB-SPH (Setup, $\mathsf{UserKeyGen}$, CerGen, $\overline{\mathsf{Priv}}$, $\overline{\mathsf{Pub}}$) is leakage-resilient smooth for some parameters.

Theorem 1. *Given an ϵ-smooth CB-SPH, let Ext: $Y \times S \rightarrow \{0,1\}^v$ be a average-case $(\log|Y| - l, \epsilon_{ext})$-extractor where S is a seeds set $\{0,1\}^\mu$, then the above transformation produces an l-leakage $(\epsilon + \epsilon_{ext})$-smooth CB-SPH.*

Proof. For a ϵ-smooth CB-SPH, we have

$$\mathbf{SD}((R, id, x, y), (R, id, x, y')) \leq \epsilon,$$

where R is the ensemble of (mpk, msk), $x \leftarrow B_{pk}, y \leftarrow$Priv$(x, sk_{id}, Cert_{id},)$ and $y' \leftarrow U_Y$. It implies that $\widetilde{H}_\infty(y|R, id, x) \approx \log|Y|$. In the presence of leakage, an adversary has access to at most l bits of leakage from the private key sk_{id} and the certificate $Cert_{id}$. Based on Lemma 1, $\widetilde{H}_\infty(y|(R, f(sk_{id}, Cert_{id}), id, x)) \geq \widetilde{H}_\infty(y|(R, id, x)) - l = \log|Y| - l$.

According to the definition of a $(\log|Y| - l, \epsilon_{ext})$ randomness extractor, we have

$$SD((R, id, f(dk_{id}), x, \bar{y}), (R, id, f(dk_{id}), x, \bar{y'})) \leq \epsilon + \epsilon_{ext}$$

where dk_{id} is the ensemble of $(sk_{id}, Cert_{id})$.

For $\bar{x} = (x, s)$ where s is chosen independently from $\{0,1\}^\mu$,

$$SD((R, id, f(dk_{id}), \bar{x}, \bar{y}), (R, id, f(dk_{id}), \bar{x}, \bar{y'})) \leq \epsilon + \epsilon_{ext}$$

where dk_{id} is the ensemble of $(sk_{id}, Cert_{id})$. So the transformed CB-SPH is l-leakage $(\epsilon + \epsilon_{ext})$-smooth. □

4 Leakage-Resilient Certificate-based Encryption

4.1 Definition

A certificate-based public key encryption scheme Π is defined by five algorithms [17]: Setup, CerGen, $\mathsf{UserKeyGen}$, $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$. Given M the message space, based on the structure of the CB-SPH, the description of the leakage resilient certificate-based Encryption is as follows.

- The first three algorithms are the same as the Setup, $\mathsf{UserKeyGen}$, CerGen algorithms in the CB-SPH. The following algorithms all take mpk as input.
- $\mathsf{Encrypt}(id, pk, m)$: Taking as input a message $m \in M$, a message sender runs this algorithm and return a ciphertext c.
- $\mathsf{Decrypt}(c, sk_{id}, Cert_{id})$: Taking as input the ciphertext c, the user runs this algorithm to return a message m using the user's private key sk_{id} and the certificate $Cert_{id}$.

Soundness of Decryption
For any $id \in ID$, any $m \in M$ and any other parameters produced by the above algorithms, we have

$$\Pr[m \neq m' \mid c \leftarrow \mathsf{Encrypt}(id, pk, m), m' \leftarrow$$
$$\mathsf{Decrypt}(c, sk_{id}, Cert_{id})] \leq negl(\kappa).$$

4.2 Semantic Security with Key Leakage

As widely known, there are two types of adversaries with different capabilities in CBE, called Type I and Type II respectively.

Type I Adversary: This type of adversary $\mathcal{A}_I$ simulates the uncertified user. Such adversary has the ability to substitute a public key for any user and learn at most $l(l \in N)$ bits for leaked secret information for the cryptographic primitive, but has no access to the master secret.

Type II Adversary: This type of adversary $\mathcal{A}_{II}$ acts an honest-but-curious certifier with the master key. Such adversary has the ability to obtain a certificate of every user and learn at most $l(l \in N)$ bits for leaked secret information for the cryptographic primitive, but is prohibited to replace any user's public keys.

There are also two types of adversaries in the leakage-resilient CBE. Here, we give the semantic security model of the leakage-resilient CBE. We define the semantic security game parameterized by the security parameter κ and a leakage parameter l.

Refer to the security model of [38], we present a LR-CBE security model. This model is described via IND-LR-CPA Game. We consider the security based on the game against leakage-resilient and adaptive chosen plaintext attacks (IND-LR-CPA).

IND-LR-CPA Game: The following is the interactive game between any probabilistic polynomial-time l-key-leakage adversary $\mathcal{A}$ of Type I or Type II and a challenger $\mathcal{B}$.

Setup: The challenger $\mathcal{B}$ takes as input a security parameter 1^κ and implements algorithm $\mathsf{Setup}(1^\kappa)$. It keeps master key msk secret and returns mpk to the attacker $\mathcal{A}$.

Phase 1: $\mathcal{A}$ makes queries adaptively, $\mathcal{B}$ handles as follows:

- Certificate queries $\mathsf{Cer}(id)$: (For Type I attacker only) $\mathcal{A}$ chooses an identity id and gives it to $\mathcal{B}$. $\mathcal{B}$ computes the corresponding certificate
$$Cert_{id} = \sigma_2(mpk, msk, id)$$
and sends it to $\mathcal{A}$.
- Private Key Extraction queries $\mathsf{PrK}(id)$: $\mathcal{A}$ produces an identity id and requests the corresponding private key sk_{id}. If the user id's public key has not been replaced then $\mathcal{B}$ responds with the user private key $sk_{id} = \sigma_1(mpk, id)$. If $\mathcal{A}$ has already replaced the user id's public key, then $\mathcal{B}$ does not provide the corresponding private key to $\mathcal{A}$.
- Request Public Key queries $\mathsf{PK}(id)$: $\mathcal{A}$ produces an identity id to $\mathcal{B}$ and requests id's public key. $\mathcal{B}$ responds by returning the public key pk for the user id computing $sk_{id} = \sigma_1(mpk, id)$ and $pk = \alpha(mpk, id, sk_{id})$.
- Public Key Replacement $\mathsf{PKR}(id)$: (For Type I attacker only)$\mathcal{A}$ can repeatedly replace the public key pk the corresponding to the user identity id with any value pk' of $\mathcal{A}$'s choice.
- Leakage queries $\mathsf{LK}(id, \mathsf{Leakage}(dk_{id})$: $\mathcal{A}$ produces an identity id and requests the corresponding the leakage information of its decryption key dk_{id} where $dk_{id} = (sk_{id}, Cert_{id})$. For any randomized function $f(\cdot)$ with l-bit output, $\mathcal{B}$ returns $f(dk_{id})$. The only restriction is that all of the leakage information about dk_{id} is l bits. For the details it is divided into two aspects as follows.
For Type I attacker $\mathcal{A}_I$, due to his ability of replacing the public key, can get the correlated leakage information $Cert'_{id}$ about the user's certificate $Cert_{id}$ besides knowing the private key sk_{id} where the length amount of $Cert'_{id}$ and sk_{id} is at most l-bit.
For Type II attacker $\mathcal{A}_{II}$, who has the master secret key msk, can get the leakage information sk'_{id} of the secret value sk_{id} besides knowing the certificate $Cert_{id}$ where the length amount of sk'_{id} and $Cert_{id}$ is at most l bits.

Challenge Stage: The adversary $\mathcal{A}$ selects an arbitrary challenge identity $id^* \in ID$ which appeared in at most l-bit leakage query. $\mathcal{A}$ also selects two equal-length messages $m_0, m_1 \in M$. The challenger $\mathcal{B}$ chooses $\beta \leftarrow \{0,1\}$ randomly, computes $c \leftarrow$ Encrypt(id^*, pk, m_b) and sends it to the adversary $\mathcal{A}$.

Phase 2: $\mathcal{A}$ adaptively makes a new sequence of queries with $id \neq id^* \in ID$ adaptively as in phase 1 except that the adversary cannot perform the leakage query.

Output: The adversary $\mathcal{A}$ outputs a bit $\beta' \in \{0,1\}$. We say that $\mathcal{A}$ wins the game if $\beta' = \beta$.

We define the advantage of $\mathcal{A}$ in the semantic security game with l-bit key-leakage to be
$$\mathrm{Adv}_{\mathcal{A},\Pi}^{LR-CPA}(\kappa, l) = |\Pr[\mathcal{A} \text{ wins}] - \tfrac{1}{2}|.$$

Definition 11. *(Leakage-resilient CBE) A CBE scheme is l-leakage-resilient if*

1) *It satisfies the soundness of decryption;*
2) *The advantage of any PPT adversary $\mathcal{A}$ in the semantic security game with l-bit key leakage is*
$$\mathrm{Adv}_{\Pi,\mathcal{A}}^{LR-CPA}(\kappa, l) = negl(\kappa);$$
3) *The relative leakage ratio of the scheme is defined to be $\alpha = \frac{l}{|dk_{id}|}$ where $|dk_{id}|$ denotes the bit size of the full decryption key sk_{id} and $Cert_{id}$.*

4.3 Construction of Leakage-Resilient CBE

It is almost natural that constructing leakage resilient CBE from l-leakage smooth CB-SPH $(\mathsf{Setup}, \mathsf{CerGen}, \mathsf{UserKeyGen}, \overline{\mathsf{Priv}}, \overline{\mathsf{Pub}})$.

Given an l-leakage smooth CB-SPH, we can construct an l-leakage resilient CBE with an identity set ID and a message set M by using the hashing value as a one-time-pad to encrypt a message directly. Recall that a CBE scheme consists of PPT algorithms $(\mathsf{Setup}, \mathsf{CerGen}, \mathsf{UserKeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$. The syntax of the first three steps is the same as that in the leakage smooth CB-SPH, and $\mathsf{Encrypt}$, $\mathsf{Decrypt}$ have the following syntax:

- $\mathsf{Encrypt}(pk, id, m)$: Compute $(x, w, d) \leftarrow \overline{\mathsf{SampA}}(pk)$, $y \leftarrow \overline{\mathsf{Pub}}(pk, id, x, w)$ and set $z = y \oplus m$. Output $c = (x, z)$.
- $\mathsf{Decrypt}(c, sk_{id}, Cert_{id})$: Parse $c = (x, z)$. Compute $y \leftarrow \overline{\mathsf{Priv}}(x, sk_{id}, Cert_{id})$. Output $m = z \oplus y$.

Theorem 2. *Given an l-leakage smooth CB-SPH, then the above construction comes to an l-leakage resilient CBE.*

Proof. For the security analysis, we proceed via a sequence of indistinguishable games. We start with Game 0 as in the real experiment and end up with a game where the view of $\mathcal{A}$ is statistically independent of the challenge bit b:

Game 0: The first game is the semantic security game with l-bit key-leakage. In the challenge stage of Game 0, the adversary selects two length-equal messages $m_0, m_1 \in M$ and a challenge identity $id^* \in ID$, the challenger chooses $b \leftarrow \{0, 1\}$ and computes $c \leftarrow \mathsf{Encrypt}(id^*, m_b)$ which we parse as $c = (x, z)$ where

$$(x, w, d) \leftarrow \overline{\mathsf{SampA}}(pk),\ y \leftarrow \overline{\mathsf{Pub}}(id^*, x, w) \text{ and set } z = y \oplus m_b.$$

Game 1: We modify the challenge stage of Game 0 as following:

$$(x, w, d) \leftarrow \overline{\mathsf{SampA}}(pk),\ \tilde{y} \leftarrow \overline{\mathsf{Priv}}(x, sk_{id^*}, Cert_{id^*}) \text{ and set } \tilde{z} = \tilde{y} \oplus m_b.$$

For the *projective property* of CB-SPH, we have $\tilde{y} = y$ with non-negligible probability. We claim that Game 0 and Game 1 are statistically indistinguishable.

Game 2: In the challenge stage of Game 2, we modify the challenge process by using a dishonest encapsulation algorithm to compute the ciphertext $c = (x, z)$ where

$$x \leftarrow \overline{\mathsf{SampB}}(pk), \tilde{y} \leftarrow \overline{\mathsf{Priv}}(x, sk_{id^*}, Cert_{id^*}), \tilde{z} = \tilde{y} \oplus m_b.$$

Game 1 and Game 2 are computationally indistinguishable due to *the hardness of the SMP* of the CB-SPH.

Game 3: The challenge ciphertext $c = (x, z)$ is computed by

$$x \leftarrow \overline{\mathsf{SampB}}(pk), \tilde{z} \leftarrow U_Y.$$

We claim that Game 2 and Game 3 are statistically indistinguishable due to the *smooth property* of the CB-SPH.

In general, Game 0 and Game 3 are indistinguishable for any PPT adversary. Obviously, the advantage of any adversary in Game 3 is negligible in κ. Therefore, the advantage of any PPT adversary in Game 0 is negligible in κ. □

5 Instantiations of CB-SPH

In this section, we present two instantiations of CB-SPH from the standard DBDH assumption and the DLWE assumption respectively.

5.1 CB-SPH Based on the DBDH Assumption

Briefly we recall the instance description DDP on the DBDH assumption which will be embedded into the concrete CB-SPH instantiation.

5.1.1 DDP Based on the DBDH Assumption

Let D be a distribution distinguish problem based on the DBDH assumption. It includes:

SampDDP(κ): Run the bilinear group algorithm $\mathsf{BLGroupGen}(\kappa)$ to generate $\text{PP} = (e, p, G, G_1)$ where G, G_1 are both p-prime order groups and $e : G \times G \to G_1$, choose $g, g_2, h \leftarrow_R G$, sets $pk = (e, G, G_1, g, g_2, h)$; Outputs an instance description $\Gamma = (X, W, PK, A_{pk}, B_{pk}, R_{pk})$ of D where $X = G \times G_1, W = Z_p, R_{pk} = \{(x, t) \in X \times W : ((g^t, e(g_2, h)^t), t)\}$, two collections of distributions A_{pk} and B_{pk} are defined by SampA and SampB as follow:

SampA(pk): Pick $t \leftarrow_R Z_p^*$, output $x = (g^t, e(g_2, h)^t) \leftarrow A_{pk}$ and $t \in W$.

SampB(pk): Pick $t, t' \leftarrow_R Z_p^*$, output $x = (g^t, e(g_2, h)^{t'}) \leftarrow B_{pk}$.

In the certificate-based settings we need some more public parameters, just like the master public key mpk, the user's identity id and the public key pk. The extensive details are described as below.

SampDDP(κ): Run bilinear group algorithm $\mathsf{BLGroupGen}(\kappa)$ to generate $PP = (e, p, G, G_1)$. Choose $g, g_1, g_2, h \leftarrow_R G$, where $g_1 = g^a$ ($a \in Z_p^*$), sets $mpk = (e, G, G_1, g, g_1, g_2, h), msk = a$; Outputs an instance description

$$\Gamma = (X, W, MPK, ID, PK, A_{pk}, B_{pk}, R_{pk})$$

of D where $X = (G \times G_1)^2, W = (Z_p)^2, R_{pk} = \{(x, (s, t)) \in X \times W : ((g_1^s g^{-s\cdot id}, e(g, g)^s, g^t, e(g_2, h)^t), w = (s, t))\}$.

SampA(mpk, id, pk): Pick $s, t \leftarrow_R Z_p^* (s \neq t)$, and output

$$x = (g_1^s g^{-s\cdot id}, e(g, g)^s, g^t, e(g_2, h)^t) \leftarrow A_{pk}$$

and $(s, t) \in W$.

SampB(mpk, id, pk): Pick $s, t, t' \leftarrow_R Z_p^* (t \neq t')$, and output

$$x = (g_1^s g^{-s\cdot id}, e(g, g)^s, g^t, e(g_2, h)^{t'}) \leftarrow B_{pk}.$$

5.1.2 Projective Hash Function

Let $\mathsf{H} = (H, SK, CERT, PK, X, A_{pk}, Y, \alpha)$ be a corresponding projective hash function, where $SK = Z_p^2, CERT = Z_p \times G, Y = G_1$. For $sk_{id} = (x, y), Cert_{id} = (Cert_1, Cert_2)$ and $x = (c_0, c_1, c_2, c_3)$, H is defined as $H_{sk_{id}, Cert_{id}}(x) = e(c_0, Cert_2)c_1^{Cert_1} e(c_2, h)^y c_3^x$.

5.1.3 DBDH-based CB-SPH

Let P be a CB-SPH for D asscociating H, which includes the following algorithms:

Setup(κ): The CA runs $\mathsf{SampDDP}(\kappa)$ to generate the master public keys $mpk = (e, p, G, G_1, g, g_1, g_2, h)$ where $g_1 = g^a \in G$ for a random number $a \in Z_p^*$. mpk will be used in the following algorithms. The master secret key is $msk = a$.

CerGen(id, msk): The CA picks $Cert_1 \in Z_p^*$ randomly and computes

$$Cert_2 = (hg^{-Cert_1})^{\frac{1}{a - id}}.$$

Then the CA returns $Cert_{id} = (Cert_1, Cert_2)$ as the user's certificate and send it to the user.

UserKeyGen(id): The user picks $x, y \in Z_p^*$ at random and gets the user private key $sk_{id} = (x, y)$. Then the corresponding public key is $pk = \alpha(sk_{id}) = {g_2}^x g^y$.

Pub(id, pk, x, w): Choose $x = (c_0, c_1, c_2, c_3) \leftarrow \mathsf{SampA}(mpk, id, pk)$ and $w = (s, t) \in W$ where $s, t \in Z_p^*$ and $s \neq t$ and compute

$$y = e(g, h)^s \cdot e(h, pk)^t.$$

Priv($x, sk_{id}, Cert_{id}$): According to $x = (c_0, c_1, c_2, c_3) \leftarrow \mathsf{SampA}(mpk, id, pk)$, with the help of the user's private key $sk_{id} = (x, y)$ and certificate $Cert_{id} = (Cert_1, Cert_2)$, compute

$$y = e(c_0, Cert_2)c_1^{Cert_1} e(c_2, h)^y c_3^x.$$

5.1.4 Remark

Our proposed CB-SPH can be transferred to a LR-CBE scheme through the way explained in Section 4. In a CBE scheme, a Type I attacker is allowed to know the private but no information of the certificate of the target identity, and a Type II attacker only knows the user's certificate of the target identity without any information about that user's private key. From our construction of CB-SPH and the use of efficient random extractors, we allow a Type I attacker finds out some secret information of the certificate or a Type II attacker gets some sensitive information of the user's private key. Even armed with such additional leakage information, the adversaries still get negligible advantage in attacking our proposed LR-CBE scheme. Obviously, for any type of adversary, the length of the relative key-leakage of our proposed LR-CBE instantiation is at most $3 \log p (= 2 \log p + \log p)$ by Lemma 2. Thus the relative leakage ratio of the LR-CBE scheme reaches $\frac{3}{4}$ maximally due to $|sk_{id}| + |Cert_{id}| = 4 \log p$.

On the other hand, based on this concrete instantiation we can present a CCA-secure leakage resilient CBE scheme. As we all know, it is usually performed by applying a suitable authentication with encryption. Borrowing the idea of [35], the scheme from the CB-SPH plus the one-time lossy filter (OT-LF) can achieve both the CCA-secure and the best leakage rate.

5.2 CB-SPH Based on the DLWE Assumption

In this section, we use that IBE scheme that was given by Gentry *et al.* [20] to construct a CB-SPH based on the DLWE assumption. It is the first cetificate-based cryprographic structure which can be transferred into a certificete-based encryption. We recall the DLWE-based DDP firstly.

5.2.1 DLWE-based DDP

Let D be a distribution distinguish problem based on the DLWE assumption.

SampDDP(κ): According to the trapdoor generation algorithm $\mathsf{TrapGen}(p, \kappa)$ of [20], it generates $A \in Z_p^{n\times m}$ along with a trapdoor $T \subset \Lambda^{\perp}(A, p)$ such that $\|\widetilde{T}\| \leq O(\sqrt{\kappa logp})$ where $\Lambda^{\perp}(A, p)$ is a set of $\{e \in Z^m$ s.t.$A^T e = \mathbf{0} \mod p\}$. Set $pk = A, sk = T$; Output an instance description $\Gamma = (X, W, PK, A_{pk}, B_{pk}, R_{pk})$ of D, where $X = Z_p^n \times Z_p, W = Z_p^n, PK = Z_p^{n\times m}, R_{pk} = \{((p, v), w) \in X \times W : ((A^T w + t, v), w)$

where error term $t \in \chi^m, v \in Z_p\}$ where χ^m is a noise distribution.

SampA(pk): Pick $w \leftarrow_R Z_p^n, t \leftarrow_R \chi^m, v \leftarrow_R Z_p$, compute $p = A^T w + t$, output $x = (p, v)$ and $w \in W$.

SampB(pk): Pick $p \leftarrow_R Z_p^m$ and $v \leftarrow_R Z_p$, output $x = (p, v)$.

In the certificate-based settings we need some more public parameters, just like master public key mpk, the user's identity id and public key pk. The extensive details are described as below.

SampDDP(κ): It generates $A \in Z_p^{n \times m}$ along with a trapdoor $T \subset \Lambda^{\perp}(A, p)$ according to the trapdoor generation algorithm $\mathsf{TrapGen}(p, \kappa)$ of [20] such that $\|\widetilde{T}\| \leq O(\sqrt{\kappa logp})$ where $\Lambda^{\perp}(A, p)$ is a set of $\{e \in Z^m \text{ s.t.} A^T e = \mathbf{0} \mod p\}$. Trapdoor function $f_A(x) = Ax \mod p$. Set $mpk = (A, f_A), msk = T$ and the user's public key is $pk = Q_{id} \in_R Z_p^{n \times m}$; Outputs an instance description $\Gamma = (X, W, MPK, PK, A_{pk}, B_{pk}, R_{pk})$ of D, where $X = (Z_p^n)^2 \times Z_p, W = Z_p^n, PK = Z_p^{n \times m}, R_{pk} = \{((p_1, p_2, v), w) \in X \times W : ((Q_{id}^T w + t_1, A^T w + t_2, v), w)$ where error terms $t_1, t_2 \in \chi^m, v \in Z_p\}$.

SampA(mpk, id, pk): Pick $w \leftarrow_R Z_p^n, t_1, t_2 \leftarrow_R \chi^m, v \leftarrow Z_p$, compute $p_1 = Q_{id}^T w + t_1, p_2 = A^T w + t_2$, output $x = (p_1, p_2, v)$ and $w \in W$.

SampB(mpk, id, pk): Pick $p_1, p_2 \leftarrow_R Z_p^m$ and $v \leftarrow_R Z_p$, output $x = (p_1, p_2, v)$.

5.2.2 Projective Hash Function

Let $\mathsf{H} = (H, SK, CERT, PK, X, A_{pk}, Y, \alpha)$ be a corresponding projective hash function, where $SK = Z_p^m, CERT = Z_p^m, Y = Z_2$. For $sk_{id} = e_{id}, Cert_{id} = t_{id}$ and $x = (p_1, p_2, v)$, H is defined as $H_{sk_{id}, Cert_{id}}(x) = y$ as $y = 1$ if $|v - (sk_{id}, Cert_{id})^T \cdot (p_1, p_2)| \leq \frac{p-1}{4}$ and $y = 0$ otherwise.

5.2.3 DLWE-based CB-SPH

Before we introduce the DLWE-based CB-SPH structure we recall some important lemma and algorithms which will be used in the CB-SPH.

We say that a matrix $A \in Z^{m \times m}$ is Z_p-invertible if A mod p is invertible as a matrix in $Z^{m \times m}$.

Lemma 3. *[4] Let $p > 2$ and a matrix $A \in Z_p^{n \times m}, m > n$. Let T be a basis for $\Lambda^{\perp}(A, p), \sigma \geq \|\widetilde{T}\| \cdot \omega(\sqrt{logm})$. Then for $u \in Z_p^n$, there is a polynomial-time algorithm $\mathsf{SamplePre}(A, T, u, \sigma)$ that returns $x \in \Lambda^u(A, p)$ sampled from a distribution statistically close to $D_{\Lambda^u(A,p),\sigma}$ where $\Lambda^u(A, p)$ is a set of $\{e \in Z^m \text{ s.t.} A^T e = u \mod p\}$.*

Algorithm [2] Sample $S(1^m)$: Let $\sigma_s = O(\sqrt{\kappa logq} \cdot \omega(logm) \cdot \sqrt{m})$.

- Let T_0 be the canonical basis of the lattice Z^m;
- For $i = 1, 2, \cdots, m$, do $s_i \leftarrow \mathsf{SampleGaussian}(Z^m, T_0, \sigma_s, \mathbf{0})$ uniformly;
- If S is Z_p-invertible, output S; otherwise repeat Step 2.

Let P be a CB-SPH for D associating H, which includes the following algorithms:

Setup(κ): Run $\mathsf{SampDDP}(\kappa)$ to generate the master public keys $mpk = (A, f_A)$ and the master secret key is $msk = T$. Let $H_1 : \{0,1\}^* \rightarrow Z_p^n$.

CerGen(mpk, id, msk): On input identity $id \in \{0,1\}^*$ and master public key A. Let $u = H_1(id) \in Z_p^n$ using the PPT algorithm $\mathsf{SamplePre}(A, T, u, \sigma)$ with trapdoor T to sample $t_{id} \leftarrow f_A^{-1}(u)$ such that $\|t_{id}\| \leq \sigma\sqrt{m}$. The CA returns $Cert_{id} = t_{id}$ as the user's certificate.

UserKeyGen(mpk, id): On input the identity id, then use the algorithm Sample $S(1^m)$ to generate a Z_p-invertible matrix S_{id}, compute $S_{id}Cert_{id} = e_{id}$. Then the user's private key is $sk_{id} = e_{id}$.

Choose a random matrix $Q_{id} \in Z_p^{n \times m}$ and compute $u_1 = Q_{id} sk_{id} \bmod p \in Z_p^{n \times m}$. The user computes the corresponding public key $pk = \alpha(sk_{id}) = (Q_{id}, u_1, u)$.

Pub(id, pk, x, w): Choose $x = (p_1, p_2, v) \leftarrow \mathsf{SampA}(mpk, id, pk)$, if $|v - (u_1 + u)^T \cdot w| \leq \frac{p-1}{4}$ then set $y = 1$ else set $y = 0$.

Priv$(x, sk_{id}, Cert_{id})$: Choose $x = (p_1, p_2, v) \leftarrow \mathsf{SampA}(mpk, id, pk)$, if $|v - (sk_{id}, Cert_{id})^T \cdot (p_1, p_2)| \leq \frac{p-1}{4}$ then set $y = 1$ else set $y = 0$.

5.2.4 Remark

In this subsection, we focus on how to use the IBE scheme [19] to construct a CB-SHP structure. We introduce a Z_p-invertible matrix $S_{id} \in Z_p^{m \times m}$ as a secret value and store it. It has two properties: 1)its norm is small; 2) its distribution is statistically close to a Gaussian distribution. The certificate $Cert_{id}$ is extracted from a distribution statistically close to a discrete Gaussian distribution by a preimage sampleable function with the master private key T which norm is also norm. For $sk_{id} = S_{id}Cert_{id}$ with the properties of S_{id} and $Cert_{id}$, it is achieved that sk_{id}'s distribution is statistically close to a Gaussian distribution and its norm is also small.

The cryptosystem based on lattice is leakage resilient in character [3], therefore the CBE scheme from the proposed DLWE-based CB-SPH is also leakage resilient for any kind of adversary and the random extractor may be unnecessary in the structure. On the other hand, based on our instantiation we can present CCA-secure CBE against key leakage attack in the random oracle.

6 Conclusion and Future Work

In this paper we presented the new notion of certificate-based smooth projecitve hashing and introduced its applications in leakage resilient encryption. We gave the formal definition of CB-SPH and showed how to transfer CB-SPH to leakage resilient one and further showed how to achieve leakage resilient certificate-based encryption (LR-CBE) schemes. With two concrete CB-SPHs, we put forward the first practical realization of LR-CBE which is based on the DBDH assumption in the standard model and presented a lattice-based CB-SPH under the DLWE assumption in the random oracle. Besides applications in the construction of LR-CBE schemes, we thought the concept of CB-SPH is of independent interest and may have other applications in the study of certificate-based cryptography.

Acknowledgments

This research is supported partially by the National Natural Science Foundation of China under Grants No. 61702259, 61672289 and 61672010, Jiangsu Government Scholarship for Overseas Studies, Postdoctoral Science Foundation of Jiangsu Province (No.1601008A), the Natural Science Fund for Colleges and Universities of Jiangsu Province (No.16KJB520018), the Social Science Fund for Colleges and Universities of Jiangsu Province (No. 2017SJB0201).

References

[1] M. Abdalla, F. Benhamouda, D. Pointcheval, "Disjunctions for smooth projective hashings: New constructions and applications," in *Advances in Cryptology (EUROCRYPT'15)* pp. 69-100, 2015.

[2] S. Agrawal, D. Boneh, X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Advances in Cryptology (CRYPTO'10)*, pp. 98-115, 2010.

[3] A. Akavia, S. Goldwasser, V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks," in *Proceedings of the 6th Theory of Cryptography Conference (TCC'09)*, pp. 474-495, 2009.

[4] J. Alwen, C.Peiker, "Generating shorter bases for hard random lattices," *Theory of Computing Systems*, vol. 48, no. 3, pp. 535-553, 2011.

[5] J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, D. Wichs, "Public-key encryption in the bounded-retrieval model," in *Advances in Cryptology (EUROCRYPT'10)*, pp. 113-134, 2010.

[6] F. Benhamouda, O. Blazy, C. Chevalier, "New techniques for SPHFs and efficient one-round PAKE protocols," in *Advances in Cryptology (CRYPTO'13)*, pp. 449-475, 2013.

[7] D. Boneh, C. Gentry, M. Hamburg, "Space-efficient identity based encryption without pairings," in *Foundation of Computer Science (FOCS'07)*, pp. 647-657, 2007.

[8] Z. Brakerski, S. Goldwasser, "Circular and leakage resilient public-key encryption under subgroup indistinguishability (or: quadratic residuosity strikes back)," in *Advances in Cryptology (CRYPTO'10)*, pp. 1-20, 2010.

[9] Y. Chen, Z. Zhang, D. Lin, et al, "Generalized (identity-based) hash proof system and its applications," *Security and Communication Networks,* vol. 9, no. 12, pp. 1698-1716, 2016.

[10] S. Chow, Y. Dodis, Y. Rouselakis, B. Waters, "Practical leakage-resilient identity-based encryption from simple assumptions," in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS'10)*, pp. 152-161, 2010.

[11] J. S. Coron, "A variant of Boneh-Franklin IBE with a tight reduction in the random oracle model," *Design, Codes Cryptography*, Vol.50, no.1, pp. 115-133, 2009.

[12] R. Cramer, V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *Advances in Cryptology (EUROCRYPT'02)*, pp. 45-64, 2002.

[13] Y. Dodis, K. Haralambiev, A. LOpez-Alt, D. Wichs, "Efficient public-key cryptography in the presence of key leakage," in *Advances in Cryptology (ASCIACRYPT'10)*, pp. 613-631, 2010.

[14] Y. Dodis, R. Ostrovsky, L. Reyzin, A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97-139, 2008.

[15] S. Dziembowski, S. Faust, "Leakage-resilient cryptography from the inner-product extractor," in *Advances in Cryptology (ASCIACRYPT'11)*, pp. 702-721, 2011.

[16] D. Galindo, P. Morillo, C. Rfols, "Improved certificate-based encryption in the standard model," *Journal of System Software*, vol. 81, no. 7, pp. 1218-1226, 2008.

[17] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Advances in Cryptology (EUROCRYPT'03)*, pp. 272-293, 2003.

[18] C. Gentry, "Practical identity-based encryption without random oracles," in *Advances in Cryptology (EUROCRYPT'06)*, pp. 445-464, 2006.

[19] C. Gentry, C. Peikert, V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th annual ACM symposium on Theory of computing (STOC'08)*, pp. 197-206, 2008.

[20] C. Gentry, C. Peikert, V. Vaikuntanathan, "How to use a short basis: Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th Annual ACM Sysmposium on Theory of Computing*, pp. 197-206, 2008.

[21] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, E. W. Felten, "Lest we remember: Cold boot attacks on encryption keys," in *Proceedings of the 17th USENIX Security Symposium*, pp. 45-60, 2008.

[22] E. Kiltz, K. Pietrzak, "Leakage resilient ElGamal encryption," in *Advances in Cryptology (ASIACRYPT'10)*, pp. 595-612, 2010.

[23] K. Kurosawa, R. Nojima, L.T.Phong, "New leakage resilient CCA-secure public key encryption," *Journal of Mathematical Cryptology*, vol. 7, no. 4, pp. 297-312, 2013.

[24] A. Lewko, Y. Rouselakis, B. Waters, "Achieving leakage resilience through dual system encryption," in *Conference on Theory of Cryptography*, pp. 70-88, 2011.

[25] J. Li, Y. Guo, Q. Yu, Y. Lu, Y. Zhang, F. Zhang, "Continuous leakage-resilient certificate-based encryption," *Information Sciences*, vol. 355, pp. 1-14, 2016.

[26] S. Li, Y. Mu, M. Zhang, F. Zhang, "Updatable lossy trapdoor functions and its application in continuous leakage," in *The 9th International Conference on Provable Security (ProvSec'16)*, pp. 309-319, 2016.

[27] S. Li, F. Zhang, "Leakage-resilient identity-based encryption scheme," *International Journal of Grid and Utility Computing,* vol. 4, no. 2/3, pp. 187-196, 2013.

[28] S. Li, F. Zhang, Y. Sun, L. Shen, "Efficient leakage resilient public key encryption from DDH assumption," *Cluster Computing*, vol. 16, pp. 797-806, 2013.

[29] J. K. Liu, J. Zhou, "Efficient certificate-based encryption in the standard model," *Security and Cryptography for Networks*, vol. 5229, pp. 144-155, 2008.

[30] S. Liu, J. Weng, Y. Zhao, "Efficient public key cryptosystem resilient to key leakage chosen ciphertext attacks," in *CT-RSA'13*, pp. 84–100, 2013.

[31] Y. Lu, J. Li, "Efficient certificate-based encryption scheme secure against key replacement attacks in the standard model," *Journal of Information Science Engineer* vol. 30, no. 5, pp. 1553-1568, (2014).

[32] M. Naor, G. Segev, "Public-key cryptosystems resilient to key leakage," in *Advances in Cryptology (CRYPTO'09)*, pp. 18-35, 2009.

[33] M. H. Nguyen, K. Tanaka, K. Yasunaga, "Leakage-resilience of stateless/stateful public-key encryption from hash proofs,", in *Australasian Conference on Information Security and Privacy (ACISP'12)*, pp. 208-222, 2012.

[34] N. Nisan, D. Zuckerman, "Randomness is linear in space," *Journal of Computer System Science*, vol. 52, no. 1, pp. 43-52, 1996.

[35] B. Qin, S. Liu, "Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter," in *Advances in Cryptology (ASIACRYPT'13)*, pp. 381-400, 2013.

[36] E. K. Reddy, "Elliptic curve cryptosystems and side-channel," *International Journal of Network Security*, vol. 12, no. 3, pp. 151-158, 2011.

[37] R. Yang, Q. Xu, Y. Zhou, R. Zhang, C. Hu, Z. Yu, "Updatable hash proof system and its applications," in *European Symposium on Research in Computer Security (ESORICS)*, pp. 266-285, 2015.

[38] Q. Yu, J. Li, Y. Zhang, "Leakage-resilient certificate-based encryption," *Security and Communication Networks*, vol. 8, no. 18, pp. 3346-3355, 2015.

[39] Q. Yu, J. Li, Y. Zhang, W. Wu, X. Huang, Y. Xiang, "Certificate-based encryption resilient to key leakage," *Journal of Systems and Software*, vol. 116, pp. 101-112, 2015.

[40] H. Wee, "Dual projective hashing and its applications – lossy trapdoor functions and more," in *Advances in Cryptology (EUROCRYPT'12)*, pp. 246-262, 2012.

[41] H. Wee, "KDM-security via homomorphic smooth projective hashing," in *19th International Conference on the Theory and Practice of Public-Key Cryptography (PKC'16)*, pp. 159-179, 2016.

[42] W. Wu, Y. Mu, W. Susilo, X. Huang, L. Xu, "A provably secure construction of certificate-based encryption from certificateless encryption," *Computer Journal*, vol. 55, no. 10, pp. 1157-1168, (2012).

Biography

Sujuan Li is an associate professor of School of Mathematical and Physical Sciences at Nanjing Tech University. She received her Ph.D. degree in applied mathematics from Nanjing Normal University. Her research interests include cryptography and information security.

Yi Mu is a professor of School of Computer and Information Technology at University of Wollongong Australia. He received his Ph.D. degree from Australian National University. His research interests include cryptography and network security.

Mingwu Zhang is a professor of School of Computer at Hubei University of Technology. He received his Ph.D. degree in cryptography from South China Agricultural University. His research interests include cryptography and network security.

A Study of Non-Abelian Public Key Cryptography

Tzu-Chun Lin
Department of Applied Mathematics, Feng Chia University
100, Wenhwa Road, Taichung 40724, Taiwan, R.O.C.
(Email: lintc@fcu.edu.tw)

(Received Feb. 12, 2017; revised and accepted June 12, 2017)

Abstract

Nonabelian group-based public key cryptography is a relatively new and exciting research field. Rapidly increasing computing power and the futurity quantum computers [52] that have since led to, the security of public key cryptosystems in use today, will be questioned. Research in new cryptographic methods is also imperative. Research on nonabelian group-based cryptosystems will become one of contemporary research priorities. Many innovative ideas for them have been presented for the past two decades, and many corresponding problems remain to be resolved. The purpose of this paper, is to present a survey of the nonabelian group-based public key cryptosystems with the corresponding problems of security. We hope that readers can grasp the trend that is examined in this study.

Keywords: Conjugacy Search Problem; Nonabelian Groups; Public Key Cryptography

1 Introduction

The development of public key cryptography was a revolutionary concept that emerged during the twentieth century. The first published study on public key cryptography was a key agreement scheme that was described by W. Diffie and M.E. Hellman in 1976 [19]. The most common public key cryptography presently in use, such as the Diffie-Hellman cryptosystem, the RSA cryptosystem, the ElGamal cryptosystem and the elliptic curve cryptosystem are number theory based and hence depend on the structure of abelian groups. Their security depends on difficulties regarding resolving some hard problems of the number theory. For instance, the RSA algorithm depends on integer factorization problem. The Diffie-Hellman, ElGamal and ECC algorithms also depend on discrete logarithmic problems (DLP). Although there have not been any successful attacks on the above public key cryptosystems the security of public key cryptosystems in use today, will be questioned due to rapidly increasing computing power and the futurity quantum computers. In 1997 [52], P.W. Shor pointed out that there are polynomial-time algorithms for solving the factorization and discrete logarithmic problems based on abelian groups during the functions of a quantum computer. Research in new cryptographic methods is also imperative, as research on nonabelian group-based cryptosystems will be one of new research priorities. In fact, the pioneering work for nonabelian group-based public key cryptosystem was proposed by N. R. Wagner and M. R. Magyarik [61] in 1985. Their idea just is not suitable for practical applications. For nearly two decades, numerous nonabelian groups have been discussed to design efficient cryptographic systems. The most frequently discussed nonabelian settings include matrix groups, braid groups, semidirect products, logarithmic signatures and algebraic erasers.

In this paper, we give an overview of known public key cryptography designed by the above mentioned nonabelian groups. These proposed nonabelian group-based public key cryptosystems rely on either encryption-decryption or on key exchange agreement. A standard model for a public key cryptographic scheme is phrased as two parties, which are referred to as Alice and Bob. Suppose that Alice wants to send a message M to Bob. A general model of encryption scheme is the following. Alice uses the encryption map f_{k_1} to encrypt the message $C = f_{k_1}(M)$, where f_{k_1} is a one-way function and is public. After receiving the cipher C, Bob uses the corresponding decryption map g_{k_2} to decode $g_{k_2}(f_{k_1}(M)) = M$, where g_{k_2} should be known only by Bob.

Many non-abelian group-based key establishment protocols are related to the Diffie-Hellman (DH) protocol, and we therefore provide a brief description of the DH-protocol. The Diffie-Hellman (DH) protocol functions as follows: Let G be a cyclic group with a generator g. Suppose that Alice and Bob want to generate a shared secret key K. Alice then randomly selects an integer $1 < a < o(g)$ and sends $A := g^a$ to Bob. Similarly, Bob randomly selects an integer $1 < b < o(g)$ and sends $B := g^b$ to Alice. Alice computes $K = B^a$, while Bob computes $K = A^b$. The security of the DH-protocol relies on the Diffie-Hellman problem (or the Discrete Logarithmic Problem).

Problem 1. *(Diffie-Hellman Problem) Let G be a group. If $g, g^x, g^y \in G$ are known, find the value of g^{xy}.*

Problem 2. *(Discrete Logarithmic Problem) Let G be a group. If $h, g \in G$ such that $h = g^x$ and h, g are known. Find the integer x.*

Problem 3. *(Conjugacy Search Problem) Let G be a nonabelian group. Let $g, h \in G$ be known such that $h = g^x$ for some $x \in G$. Find x. Here g^x stands for $x^{-1}gx$.*

Nonabelian group-based public key cryptography is a relative new research field. In contrast to abelian groups the conjugacy search problem and its variant versions are hard problems on some nonabelian groups. The conjugacy search problem and its variant versions play an important role for the security on nonabelian group-based public key cryptography.

In this paper, we give a survey of the representative nonabelian group-based public key cryptosystems so far. Their algorithms are very different.

2 Matrix Groups

2.1 Yamamura's Encryption Scheme

At PKC'98, A. Yamamura [64] proposed a public key encryption scheme based on the modular group $SL(2, \mathbb{Z})$. It is well known that $SL(2, \mathbb{Z})$ is generated by two matrices $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ and $T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, where the orders of both generators are $o(S) = 4$ and $o(T) = \infty$ and the matrix $ST = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}$ is of order 6. Also, $SL(2, \mathbb{Z})$ is generated by the matrices S and ST subject to the relations $S^4 = (ST)^4 = I$ and $(ST)^3 = S^2$. For a matrix $N \in SL(2, \mathbb{Z})$, the matrices $A := N^{-1} \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix} N$ and $B := N^{-1} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} N$ satisfy the relations $A^6 = B^4 = I$ and $A^3 = B^2$. Therefore, the matrices A and B generate $SL(2, \mathbb{Z})$.

1) Key Generation: Bob

   a. chooses two matrices $V_1 := (BA)^i$ and $V_2 := (BA^2)^j \in SL(2, \mathbb{Z})$ for some $i, j \in \mathbb{N}$.

   b. chooses matrices $M \in GL_2(\mathbb{C})$ and $F_1(X), F_2(X) \in Mat_2(\mathbb{C}[X])$ and $a \in \mathbb{C}$ such that $F_1(a) = V_1$ and $F_2(a) = V_2$.

   c. computes $W_1(X) := M^{-1}F_1(X)M$ and $W_2(X) := M^{-1}F_2(X)M$.

   d. Bob's public key: $W_1(X), W_2(X)$.
   Bob's private key: M, a.

2) Encryption: Let $b_1 \cdots b_n \in \{0, 1\}^n$ be the message. Alice computes the ciphertext

$$C(X) := W_2(X) \prod_{i=1}^{n} (W_1(X)^{b_i+1} W_2(X)).$$

3) Decryption: From the ciphertext $C(X)$ and Bob's private key (M, a) the message $b_1 \cdots b_n$ can be recovered by means of a procedure described in [64].

4) Security Analysis:
The protocol is based on conjugacy search problem and root problems. But, R. Steinwandt [55] in 1992) pointed out that the Yamamura's Encryption Scheme is insecure. Suppose that an adversary Eve intercepted to the cipher $C(X)$. She can compute

$$D(X) := W_2(X)^{-1}C(X) = \prod_{i=1}^{n} (W_1(X)^{b_i+1} W_2(X)).$$

The entries of the matrix

$$((W_1(X)^{b_i+1} W_2(X))^{-1} D(X)$$

should be polynomials over $\mathbb{C}$. Beginning with the first bit b_1, if at least one of the entries of $D_1 := ((W_1(X)^2 W_2(X))^{-1} D(X)$ involves a nonconstant denominator then we can conclude $b_1 = 0$; otherwise $b_1 = 1$. Similarly, if the matrix $D_2 := ((W_1(X)^2 W_2(X))^{-1} D_1(X)$ contains a nonpolynomial entry then we can conclude $b_2 = 0$; otherwise $b_2 = 1$. The process continues until all bits $b_i, i = 1, \cdots, n$ are recovered. This means that the plaintext $b_1 \cdots b_n \in \{0, 1\}^n$ can be recovered efficiently from ciphertext $C(x)$ and the public data alone.

2.2 Two Rososhek-Matrix Cryptosystems

In 2013, S. K. Rososhek [49] proposed a ElGamal-like encryption scheme -called BMMC ((Basic Matrix Modular Cryptosystem) - by using matrices over $\mathbb{Z}_n$.

1) BMMC: Let n be a large positive integer and let $G(\alpha, \beta, \gamma)$ be a free subgroup of the general linear group $\mathrm{GL}(2, \mathbb{Z}_n)$ generated by three generators A, B and C, where $\alpha, \beta, \gamma \in \mathbb{Z}$ with $\mid \alpha \mid, \mid \beta \mid, \mid \gamma \mid \geq 3$, $A = \begin{pmatrix} 1 & 0 \\ \alpha & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}$ and $C = \begin{pmatrix} 1-\gamma & r \\ -\gamma & \gamma+1 \end{pmatrix}$. Let q be the order of the group $\mathrm{GL}(2, \mathbb{Z}_n)$. All the data above is public.

   a. Key Generation: Bob

      i. chooses two random matrices P_1 and U in $G(\alpha, \beta, \gamma)$ with $P_1U \neq UP_1$ and three integers k, s, l with $-q \leq k, s \leq q$ and $2 \leq q$.

      ii. computes $P_2 := U^{-s}P_1^kU^s$ and $P_3 := U^l$.

      iii. The public key: n, P_1, P_2, P_3.
      The private key: U, k, s.

   b. Encryption: Let the message $m \in Mat(2, \mathbb{Z}_n)$ be a matrix. Alice chooses integers $r, t \in \mathbb{Z}_n$ and then computes the ciphertext

$$(C_1, C_2) := (P_3^{-r}P_1^tP_3^r, mP_3^rP_2^{-t}P_3^{-r}).$$

c. Decryption: Bob computes m by using his private key k, s:

$$C_2 U^{-s} C_1^k U^s = m.$$

d. Security Analysis:
If Eve want to break the system, Eve has to solve the transformation and hybrid problems that are more complicated than the discrete logarithm problem in the group of the same cardinality. The both hard problems are described as follows.

Problem 4. *(The Transformation Problem): Find all solutions (Z, y) of the equation $ZP_1Z^{-1} = P_1^y$, where $Z \in \mathrm{GL}(2, \mathbb{Z}_n)$ and $\mid y \mid < q$ is an integer.*

Problem 5. *(The Hybrid Problem): Find all solutions (Y, x) of the equation $Z_0 = Y^x$, where $Y \in \mathrm{GL}(2, \mathbb{Z}_n)$ and $\mid x \mid < q$ is an integer.*

2) MMMC1: The BMMC requires three matrix modular exponentiations for key generation. There are three exponentiation under encryption and two exponentiations under decryption. In order to speed the algorithm, S. K. Rososhek [50] gave two modified schemes named MMMC1 (Modified Matrix Modular Cryptosystem one) and MMMC2. The both modified schemes are similar. We only introduced the MMMC1 here.

a. Key Generation: Bob

i. computes the integer n, where n may be either a power of a prime p^r or a product $n = pq$ of two distince primes.

ii. determines two invertible matrices $V, W \in \mathrm{GL}(2, \mathbb{Z}_n)$ in order to define two commuting inner automorphisms α, β of the ring $Mat(2, \mathbb{Z}_n)$: $\alpha(D) := V^{-1}DV$ and $\beta(D) := W^{-1}DW$, for all $D \in Mat(2, \mathbb{Z}_n)$.

iii. computes two automorphisms $\phi := \alpha^2\beta$ and $\psi := \alpha\beta^2$.

iv. chooses a matrix $L \in \mathrm{GL}(2, \mathbb{Z}_n)$ such that $L \notin G$.

v. The public key: $n, \phi(L), \psi(L^{-1})$.
The private key: V, W, α, β.

b. Encryption: Let the message $m \in Mat(2, \mathbb{Z}_n)$ be a matrix. Alice

i. chooses $Y \in G$ and define an inner automorphism ζ of the ring $Mat(2, \mathbb{Z}_n)$ by $\zeta(D) := Y^{-1}DY$.

ii. computes the matrices $\zeta(\phi(L)), \zeta(\psi(L^{-1}))$ and $m\zeta(\phi(L))$.

iii. chooses a unit $\gamma \in \mathbb{Z}_n$.

iv. computes the ciphertext

$$(C_1, C_2) = (\gamma^{-1} \cdot \zeta(\psi(L^{-1}),\ \gamma \cdot m \cdot \zeta(\phi(L))).$$

c. Decryption: Bob decrypts the message using his private key

$$C_2 \cdot \alpha^{-1}\beta(C_1) = m.$$

d. Security Analysis:
The security of the scheme is based on the "random salt" conjugacy search problem. This is for the given matrices A, B in $Mat(2, \mathbb{Z}_n)$ to find an invertible matrix $X \in \mathrm{GL}(2, \mathbb{Z}_n)$ and an integer $0 < \gamma < n$ such that $X^{-1}AX = \gamma B$.
If the integer γ in the encryption algorithm is removed, then the system is insecure. This is because the usual conjugacy search problem on the general linear group $\mathrm{GL}(2, \mathbb{Z}_n)$ is not hard. The equation $C_1 = Y^{-1}\psi(L^{-1})Y$ can be transformed to a system of four linear equations with four unknowns. On the other hand, the author [50] claimed that the "salt" γ can be found only under brute force attack and for large n this problem becomes intractable.

More about public key cryptosystems based on matrices, see for example [9,21,24,27,32,49,50,55–57,64] for an example.

3 Braid Groups

The braid groups were first introduced explicitly by E. Artin in 1925 [4]. There are several ways to represent braids, but the most common is through the use of Artin generators and the fundamental braid [15]. The (Artin's) braid groups, denoted as B_n, are groups of braids on n strands defined by the following presentation

$$B_n := < \sigma_1, \cdots, \sigma_{n-1} \mid \sigma_i\sigma_{i+1} = \sigma_{i+1}\sigma_i,$$
$$\sigma_{i+1}\sigma_i\sigma_{i+1} = \sigma_i\sigma_{i+1}\sigma_i, 1 \le i \le n-1 > .$$

These are non-abelian torsion-free groups. The precise description, in particular is the geometric interpretation of Artin braid groups, see *e.g.* [2,10,17,18,30,36]. Due to their efficient computational quality, Artin's braid groups seemed to be a good candidate as a platform group for cryptographic applications.

At the beginning of the twenty-first century, some braid group-based public key cryptosystems were proposed. The pioneering papers for braid group-based cryptography include the Anshel-Anshel-Goldfeld scheme [2] in 1999 and the Ko-Lee *et al.* scheme [36] in 2000. Since then, braids group-based cryptography has attracted a great deal of attention. The security of the most proposed braid group cryptographic schemes is based on the conjugacy search problem or its variant versions, *e.g.* the membership search problem.

Problem 6. *(Membership Search Problem (or, Multiple Conjugacy Search Problem)) Given elements $x, a_1, a_2, \cdots, a_n$ of a group G, find an expression of x as a word in $a_1, a_2, \cdots, a_n$ (if it exists).*

Unfortunately, the conjugacy search problem on linear groups is not hard, and braid groups are linear groups [11,37]. Although the most proposed braid group-based cryptographic schemes are vulnerable to several announced attack methods [28], the research of the braid groups for cryptography has not decreased. Apart from the conjugacy search problem, there are other hard problems in braid groups that have not been studied extensively. We therefore present the works of Ko-Lee *et al.* and Anshel *et al.* and the corresponding attacks based on linear representations of the braid groups. The algorithms can be applied not only to braid groups, but also to any nonabelian groups.

3.1 Two Ko-Lee *et al.* Schemes

Let LB_k and RB_{n-k} be commuting subgroups of the braid group B_n, where $0 < k < n$, consisting of braids made by braiding left k strands and by braiding right $n-k$ strands among n strands, respectively. For any $a \in LB_k$ and $b \in RB_{n-k}$, the commutative rule holds: $ab = ba$.

1) The Key Agreement Scheme [36]: The algorithm is a Diffie-Hellman like algorithm.

   a. The public key: braids groups B_n, LB_k, RB_{n-k} and a braid $x \in B_n$.

   b. Alice chooses a random secret braid $a \in LB_k$ and sends $y_1 := axa^{-1}$ to Bob.
   Bob chooses a random secret braid $b \in RB_{n-k}$ and sends $y_2 := bxb^{-1}$ to Alice.

   c. Alice receives y_2 and computes the shared key $K = ay_2a^{-1}$.
   Bob receives y_2 and computes the shared key $K = by_1b^{-1}$.

2) The Encryption Scheme [36]:

   a. Bob's public key: x, y, where $x \in B_n$, $y := axa^{-1}$ and the hash function $H : B_n \to \{0,1\}^l$.
   Bob's private key: $a \in LB_k$.

   b. Encryption: Let $m \in \{0,1\}^l$ be a plaintext. Alice

      i. chooses a braid $b \in RB_{n-k}$ at random.

      ii. computes the ciphertext (c, d), where

$$c = bxb^{-1}\ , \quad d = H(byb^{-1}) \oplus m.$$

   c. Decryption: Bob uses the prime key a to recover the message

$$m = H(aca^{-1}) \oplus d.$$

3) Security Analysis:
   The security of both of these schemes is based on the conjugacy search problem on braid groups. To break the both schemes, it suffices for Eve to solve the Braid Diffie-Hellman Conjugacy Problem.

Problem 7. *(Diffie-Hellman Conjugacy Problem) Let A and B be commuting subgroups of a group G with $[A, B] = 1$, and let $g \in G$ be given. Given a pair (g^a, g^b) with $a \in A$ and $b \in B$, find g^{ab}.*

The Ko-Lee *et al.* key agreement scheme can be attacked by using the Lawrence-Krammer representation

$$\mathcal{K} : B_n \longrightarrow \mathrm{GL}(\frac{n(n-1)}{2}, \mathbb{Z}[t^{\pm 1}, \frac{1}{2}]).$$

The proposed algorithm to solve the braid DH problem is described roughly as follows. Suppose that Eve can find a matrix A such that

$$\begin{aligned} \mathcal{K}(y_1)A &= A\mathcal{K}(y_1), \\ \mathcal{K}(\sigma_i)A &= A\mathcal{K}(\sigma_i), \end{aligned}$$

for all generators $\sigma_i \in LB_k$. Then, $A\mathcal{K}(y_2)A^{-1} = A\mathcal{K}(b)\mathcal{K}(x)\mathcal{K}(b)^{-1}A^{-1} = \mathcal{K}(b)\mathcal{K}(y_1)\mathcal{K}(b)^{-1} = \mathcal{K}(K)$. Note that the Lawrence-Krammer representation is faithful and one can effectively find the image $\mathcal{K}(g)$ for every $g \in B_n$. Moreover, one can effectively recover $K \in B_n$ from its image $\mathcal{K}(K)$ by using the Cheon-Jun inversion algorithm [16,59].

3.2 Anshel-Anshel-Goldfeld Scheme

In contrast to Ko-Lee *et al.* schemes the Anshel-Anshel-Goldfeld key agreement scheme [2] requires no commuting subgroups. Let G be a public nonabelian group and $a_1, \cdots, a_k, b_1, \cdots, b_m \in G$ be public.

1) The Algorithm.

   a. Alice chooses a random secret $x = x(a_1, \cdots, a_k) \in G$ as a word in $a_1, \cdots, a_k$ and sends $b_1^x, \cdots, b_m^x$ to Bob.
   Bob chooses a random secret $y = y(b_1, \cdots, b_m) \in G$ as a word in $b_1, \cdots, b_k$ and sends $a_1^y, \cdots, a_k^y$ to Alice.

   b. Alice computes $x(a_1^y, \cdots, a_k^y) = x^y = y^{-1}xy$ and $x^{-1}(y^{-1}xy) = K$.

   c. Bob computes $y(b_1^x, \cdots, b_m^x) = y^x = x^{-1}yx$ and $(y^{-1}(x^{-1}yx))^{-1} = K$.

2) Security Analysis:
   In the paper [2], braid groups are selected as platform groups for the scheme. The security of the AAG scheme is based on the multiple conjugacy search problem, which is otherwise called the membership search problem. However, for Eve to extract the shared key K out of the public information, it suffices to solve the Commutator KE Problem, which is otherwise called the Anshel-Anshel-Goldfeld Problem, in polynomial time.

Problem 8. *(Commutator Key Exchange Problem) Let G be a group. Let $a_1, \cdots, a_k, b_1, \cdots, b_k \in G$ and* *let $a \in < a_1, \cdots, a_k >$ and $b \in < b_1, \cdots, b_k >$. Given $a_1, \cdots, a_k, b_1, \cdots, b_k, a_1^b, \cdots, a_k^b, b_1^a, \cdots, b_k^a$, compute $a^{-1}b^{-1}ab$.*

Let S be a subset of $Mat(n, \mathbb{F})$. The centralizer of S is the set

$$C(S) := \{c \in Mat(n, \mathbb{F}) \mid cs = sc, \ \forall\ s \in S\}.$$

The centralizer $C(S)$ is a subspace of the vector space $Mat(n, \mathbb{F})$ over a field $\mathbb{F}$. The proposed algorithm [59] to solve the commutator key exchange problem is described roughly as follows. See [59] for details.

a. Use the method of Cheon and Jun [16] to reduce the commutator key exchange problem in matrix groups over fields.

$$B_n \overset{\mathcal{K}}{\hookrightarrow} \mathrm{GL}(\frac{n(n-1)}{2}, \mathbb{Z}[t^{\pm 1}, \frac{1}{2}]) \twoheadrightarrow \mathrm{GL}(\frac{n(n-1)}{2}, \mathbb{F}),$$

where $\mathbb{F} = \mathbb{Z}[t]/ < p, f(t) > = \mathbb{Z}[t^{\pm 1}, \frac{1}{2}]/ < p, f(t) >$ is a finite field of the order $p^{\deg f(t)}$, p is a prime and $f(t)$ is an irreducible polynomial.

b. Compute a basis for $C(C(b_1, \cdots, b_k))$.

c. Find a matrix x (and its inverse x^{-1}) that satisfies the following homogeneous system of linear equations

$$\begin{array}{rcl} b_1 \cdot x & = & x \cdot b_1^a, \\ & \vdots & \\ b_k \cdot x & = & x \cdot b_k^a. \end{array}$$

Thus, $b_i^x = b_i^a$ and $xa^{-1} \in C(b_1, \cdots, b_k)$.

d. If $y \in C(C(b_1, \cdots, b_k))$, then $(xa^{-1})y = y(xa^{-1})$ and $(xa^{-1})y^{-1} = y^{-1}(xa^{-1})$. Therefore,

$$\begin{array}{rcl} x^{-1}y^{-1}xy & = & x^{-1}y^{-1}(xa^{-1})ay \\ & = & x^{-1}(xa^{-1})y^{-1}ay \\ & = & a^{-1}a^y. \end{array}$$

e. Find a matrix $y \in C(C(b_1, \cdots, b_k))$ (and its inverse y^{-1}) that satisfies the following homogeneous system of linear equations

$$\begin{array}{rcl} a_1 \cdot y & = & y \cdot a_1^b, \\ & \vdots & \\ a_k \cdot y & = & y \cdot a_k^b. \end{array}$$

f. Let $a = a_{i_1}^{\epsilon_1} \cdots a_{i_m}^{\epsilon_m}$. Then, compute

$$\begin{array}{rcl} a^y & = & (a_{i_1}^{\epsilon_1})^y \cdots (a_{i_m}^{\epsilon_m})^y = (a_{i_1}^y)^{\epsilon_1} \cdots (a_{i_m}^y)^{\epsilon_m} \\ & = & (a_{i_1}^b)^{\epsilon_1} \cdots (a_{i_m}^b)^{\epsilon_m} = (a_{i_1}^{\epsilon_1})^b \cdots (a_{i_m}^{\epsilon_m})^b \\ & = & a^b. \end{array}$$

g. Compute $a^{-1}a^y = a^{-1}a^b = a^{-1}b^{-1}ab$.

h. Recover the shared key K from $a^{-1}b^{-1}ab$ by using the Cheon-Jun inversion algorithm.

For more about braid groups and braid group-based public key cryptosystems, see for example [2, 4, 10, 11, 15–18, 23, 28, 30, 36, 37, 39, 44, 45, 47, 53, 59].

4 Stickel's Schemes

In 2003, E. Stickel presented the algorithms based on the Diffie-Hellman type of nonabelian groups. The algorithms cover the key agreement, authentication and digital signature purposes. Let G be a finite nonabelian group and let $a, b \in G$ with $ab \neq ba$ and $o(a) = N, o(b) = M > 1$.

1) Stickel's Key Agreement Scheme

a. Alice chooses two random natural number $n < N, m < M$ and sends $u := a^n b^m$ to Bob.
Bob chooses two random natural number $r < N, s < M$ and sends $v := a^r b^s$ to Alice.

b. Alice computes the shared secret key $K = a^n v b^m$.
Bob computes the shared secret key $K = a^r u b^s$.

2) Security Analysis:
Suppose that Eve wants to break the system and she has intercepted the values u and v. In order to get the secret shared key K, Eve does not have to find a pair of integers (n, m) (or (r, s)), but to solve the decomposition search problem [45, 54].

Problem 9. *(Decomposition Search Problem) Given a recursively presented (semi)group G, two recursively generated sub(semi)groups $A, B \in G$, and two elements $u, w \in G$. Find two elements $x \in A$ and $y \in B$ such that $x \cdot w \cdot y = u$, provided at least one such pair of elements exists.*

Suppose that Eve can find a pair $x, y \in G$ which satisfies the system

$$\left\{ \begin{array}{rcl} xa & = & ax \\ yb & = & by \\ u & = & xwy \end{array} \right.$$

then Eve can use Bob's transmission v to compute

$$xvy = xa^r wb^s y = a^r xwyb^s = a^r ub^s = K.$$

3) Suggested Platforms: In the paper [56], it was suggested that the general linear group $\mathrm{GL}_k(\mathbb{F}_{2^l})$ is used as the platform group G. Then the above system of three equations including a nonlinear equation can be translated to a system of three linear equations

$$\left\{ \begin{array}{rcl} x^{-1}a & = & ax^{-1} \\ yb & = & by \\ xu & = & wy. \end{array} \right.$$

It makes also the protocol vulnerable to linear algebra attacks. However, the system is worth preserving. The author of the paper [54] suggested semigroups with a great deal of non-invertible elements, and then the linear algebra attack would not work. Whether a semigroup (with a lot of non-invertible elements) as the platform makes the protocol vulnerable to another attacks is unclear.

5 Semidirect Products

In [47] (2001), S.-H. Paeng *et al.* described a public key encryption protocol based on a semidirect product of abelian groups connecting with the inner automorphism.

Definition 5.1. Let G and H be groups and $\rho : H \longrightarrow \mathrm{Aut}(G)$ be a group homomorphism. The semidirect product $G \rtimes_\rho H$ is a nonabelian group

$$G \rtimes_\rho H := \{(g, h) \mid g \in G, h \in H\}$$

under the group operation

$$(g_1, h_1) \cdot (g_2, h_2) := (g_1 \rho(h_1)(g_2), h_1 h_2).$$

Definition 5.2. Let G be a nonabelian group. Fix an element $g \in G$. An automorphism $\mathrm{Inn}(g) : G \longrightarrow G$ defined by

$$\mathrm{Inn}(g)(x) := gxg^{-1},\ \forall\, x \in G$$

is called the inner automorphism of G by g.

Problem 10. *(Special Conjugacy Search Problem) Let G be a nonabelian group. Given an element $\mathrm{Inn}(g) \in \mathrm{Inn}(G)$. Find $g' \in G$ such that $\mathrm{Inn}(g') = \mathrm{Inn}(g)$.*

5.1 The Encryption Scheme

Let Γ be a semidirect product of the groups $\mathrm{SL}(2, \mathbb{Z}_p)$ and $\mathbb{Z}_p$. $\mathrm{Inn}(\Gamma) := \{\mathrm{Inn}(g) \mid g \in \Gamma\}$ be the inner automorphism group of Γ. The encryption scheme in [47]:

1) Bob's public key: $\mathrm{Inn}(g), \mathrm{Inn}(g^a), g := (x, y) \in \Gamma \setminus \mathcal{Z}(\Gamma)$.
   Bob's private key: $a \in \mathbb{Z}_{|\Gamma|}$.

2) Encryption: Let $m := (m_1, m_2) \in \Gamma \setminus \mathcal{Z}(\Gamma)$ with $mg \neq gm$ be a message. Alice

   a. chooses $b \in \mathbb{Z}$ and computes $\mathrm{Inn}(g^a)^b$.

   b. computes $E = \mathrm{Inn}(g^{ab})(m)$.

   c. computes $\phi = \mathrm{Inn}(g)^b$.

   Alice sends to Bob the cipher (E, ϕ).

3) Decryption: Bob computes $m = \phi^{-a}(E)$.

4) Security Analysis:
   The security of the encryption scheme is based on the difficulty of the special conjugacy search problem and the discrete logarithmic problem.

   If Eve wants to break the system, she has to find the private key a. Eve can try to find an element g_0 such that $\mathrm{Inn}(g_0) = \mathrm{Inn}(g^a)$. Then, it holds that $g_0 = g^a z$ for some $z \in \mathcal{Z}(\Gamma)$. After that Eve has to check whether $g_0 z^{-1} \in < g >$. If that is the case, then it goes back to solve the discrete logarithmic problem in the cyclic group $< g >$. Indeed, if the subgroup $\mathcal{Z}(\Gamma)$ of Γ is large, then it is less efficient to determine whether $g_0 z^{-1} \in < g >$.

   On the other hand, Eve considers directly solving the discrete logarithmic problem in the group $< \mathrm{Inn}(g) >$. The most efficient known method- the index calculus- cannot applied to the group $< \mathrm{Inn}(g) >$. In general case, the expected run times for solving he discrete logarithmic problem are $\mathcal{O}(\sqrt{p})$.

5) Suggested Platforms:
   The Author [47] employ a semi-direct product Γ of groups as the platform group of the system. Let $\Gamma = SL(2, \mathbb{Z}_p) \rtimes_\rho \mathbb{Z}_p$ an let $\rho := \mathrm{Inn} \circ \rho_1 : \mathbb{Z}_p \to \mathrm{Aut(SL)(2, \mathbb{Z}_p))}$ be the automorphism, which is a composition of an inner automorphism Inn with an isomorphism ρ_1. For the DLP to be a hard problem in $< \mathrm{Inn}(g) >$, we choose 160-bit prime p. Then the security of the system is comparable to 1024-bit RSA.

5.2 HKKS-Key Exchange Protocol

In 2013, M. Habeeb *et al.* [29] proposed a new key agreement protocol (HKKS) by using semidirect products which is very different from the S.-H. Paeng *et al.* scheme. Let G be a (semi)group and let $\Gamma = G \rtimes H$ be a semidirect product, where $H \leq \mathrm{Aut}(G)$ is a subgroup. Let $(g, \phi) \in \Gamma$ be the public key for the protocol.

1) The HKKS-Key Exchange Portocol in [29]:

   a. Alice chooses a private number $m \in \mathbb{N}$ and computes $(g, \phi)^m = (\phi^{m-1}(g) \cdots \phi^2(g)\phi(g)g,\ \phi^m)$.
   Alice sends to Bob the first component

   $$a := \phi^{m-1}(g) \cdots \phi^2(g)\phi(g)g.$$

   Bob chooses a private number $n \in \mathbb{N}$ and computes $(g, \phi)^n = (\phi^{n-1}(g) \cdots \phi^2(g)\phi(g)g,\ \phi^n)$.
   Bob sends to Alice the first component

   $$b := \phi^{n-1}(g) \cdots \phi^2(g)\phi(g)g.$$

   b. Alice chooses any $x \in H$ and computes $(b, x)(a, \phi^m) = (\phi^m(b) \cdot a,\ x \cdot \phi^m)$.
   Bob chooses any $y \in H$ and computes $(a, y)(b, \phi^n) = (\phi^n(a) \cdot b,\ y \cdot \phi^n)$.

   c. The shared secret key K of Alice and Bob is the first component of $(b, x)(a, \phi^m) = (a, y)(b, \phi^n) = (g, \phi)^{m+n}$

   $$K = \phi^n(a) \cdot b = \phi^m(b) \cdot a.$$

2) Suggested Platforms:
In this paper [29], the authors consider the semigroup G of 3×3 matrices over the group ring $\mathbb{Z}_7[A_5]$, where A_5 is the alternating group on 5 letters and an extension of G by an inner automorphism to get a platform for the protocol. The public map ϕ is defined as an inner automorphism by using a fixed invertible matrix $h \in G$

$$\phi(g) := h^{-1}gh \text{ and } \phi^k(g) := h^{-k}gh^k$$

for any matrix $g \in G$ and any integer $k \geq 1$.

In order to reduce key size and to speed up computation of the algorithm, the authors of the paper [33] consider the semigroup G of 2×2 matrices over the binary field $\mathbb{F}_{2^{127}}$ and an extension of G by an endomorphism ϕ, which is a composition of a conjugation by a matrix $h \in \mathrm{GL}(2, \mathbb{F}_{2^{127}})$ with the endomorphism ψ as the platform semigroup. The public map ϕ is thus defined as follows

$$\begin{aligned} \phi(g) &:= h^{-1}\psi(g)h, \text{ and} \\ \phi^k(g) &:= (\prod_{i=0}^{k-1} \psi^i(h^{-1})\psi^k(g)(\prod_{i=k-1}^{0} \psi^i(h)), \end{aligned}$$

for any matrix $g \in G$ and any integer $k \geq 1$. The change of G reduces the bit complexity of a public matrix g from 1620-bits into 508-bits. All computation are done by well known methods of fast computation in finite binary fields. However, the choice of the both platforms makes the protocol unable to resist a linear algebra attack and a linear decomposition attack.

3) Security Analysis:
We give the linear algebra attack in the original version. Other attacks referred to [20, 29, 33, 34, 48]. If the inner automorphism ϕ_h by a matrix h over a field, i.e. $\phi_h(x) = hxh^{-1}$, is selected as the automorphism ϕ, then the HKKS-key exchange protocol is vulnerable to the linear algebra attack. The reason is described as follows:

a. Recall that Alice sends the matrix a to Bob

$$a = (\prod_{i=m-1}^{1} h^{-i}gh^i) \cdot g = h^{-m}(hg)^m,$$

and Bob sends the matrix b to Alice

$$b = (\prod_{i=n-1}^{1} h^{-i}gh^i) \cdot g = h^{-n}(hg)^n.$$

The shared secret key K is

$$K = \phi^m(b) \cdot a = \phi^n(a) \cdot b = h^{-(m+n)}(gh)^{m+n}.$$

b. Suppose that Eve wants to break the protocol. She finds two matrices X and Y satisfying the system of two linear and one nonlinear equations

$$\left\{ \begin{aligned} Xh &= hX, \\ Y(hg) &= (hg)Y, \\ XY &= h^{-m}(hg)^m. \end{aligned} \right.$$

If the matrix X is invertible, the system can be translated to the system of three linear equations

$$\left\{ \begin{aligned} X^{-1}h &= hX^{-1}, \\ Y(hg) &= (hg)Y, \\ Y &= X^{-1}h^{-m}(hg)^m. \end{aligned} \right.$$

It makes also the protocol vulnerable to a linear algebra attack, since Eve can thus compute the shared secret key by applying the solution (X, Y)

$$X(h^{-n}(hg)^n)Y = h^{-n}(XY)(hg)^n = K.$$

The next key exchange protocol is a modified version in order to prevent the linear algebra and linear decomposition attacks.

5.3 Modified HKKS-Key Exchange Protocol [33]

Let the automorphism ϕ be the inner automorphism ϕ_h by an invertible matrix h, i.e. $\phi_h(x) = hxh^{-1}$ and let $\mathrm{Ann}(hg) := \{x \mid x \cdot (hg) = O\}$ be the annihilator of the matrix hg, where O is denoted as the zero matrix. Alice and Bob agree on public matrices g and h, where h is invertible and g is not.

1) The modified version:

a. Alice chooses a secret number $m \in \mathbb{N}$ and a secret nonzero matrix $R \in \mathrm{Ann}(hg)$, and then computes

$$\begin{aligned} (g, \phi)^m &= (\phi^{m-1}(g) \cdots \phi^2(g)\phi(g)g,\ \phi^m) \\ &= (h^{-m}(hg)^m, \phi^m) = (a,\ \phi^m). \end{aligned}$$

Alice sends to Bob the matrix $a + R$.
Bob chooses a secret number $n \in \mathbb{N}$ and a secret nonzero matrix $S \in \mathrm{Ann}(hg)$, and then computes

$$\begin{aligned} (g, \phi)^n &= (\phi^{n-1}(g) \cdots \phi^2(g)\phi(g)g,\ \phi^n) \\ &= (h^{-n}(hg)^n, \phi^m) = (b,\ \phi^n). \end{aligned}$$

Bob sends to Alice the matrix $b + S$.

b. Alice chooses any $x \in H$ and computes

$$(b + S, x)(a + R, \phi^m) = (\phi^m(b) \cdot a,\ x \cdot \phi^m).$$

Bob chooses any $y \in H$ and computes

$$(a + R, y)(b + S, \phi^n) = (\phi^n(a) \cdot b,\ y \cdot \phi^n).$$

c. The shared secret key K of Alice and Bob is then

$$K = \phi^m(b) \cdot a = \phi^n(a) \cdot b.$$

2) Security Analysis:
The protocol uses R and S to hide a and b, respectively. Here, Eve would be looking for matrices X, Y and Z of the system of four equations

$$\begin{aligned} Xh &= hX, \\ Y(hg) &= (hg)Y, \\ Z \cdot (hg) &= O, \\ XY + Z &= h^{-m}(hg)^m + R. \end{aligned}$$

The last equation is not linear. The linear algebra attack does not work against this protocol. However, it is vulnerable against the linear decomposition attack which is described as follows [20, 48].

a. First, construct a linear space W generated by all elements of the form $h^{-k}(hg)^k, k \in \mathbb{N} \cup \{0\}$, with a basis $\{e_1, \cdots, e_l\}$, where $e_i = h^{-k_i}(hg)^{k_i}$, $k_i \in \mathbb{N}$. Choose a basis $\{f_1, \cdots, f_t\}$ of the linear space $\mathrm{Ann}(hg)$ such that $\{e_1, \cdots, e_l, f_1, \cdots, f_t\}$. This consists of a basis of the space $W + \mathrm{Ann}(hg)$.

b. For public data $a+R$ and $b+S$ in $W+\mathrm{Ann}(hg)$, we can effectively find one matrix $S_1 \in \mathrm{Ann}(hg)$ and the coefficients $\eta_i, \nu_j \in \mathbb{F}$ such that

$$\begin{aligned} b + S &= h^{-n}(hg)^n + S \\ &= \sum_{i=1}^{l} \eta_i h^{-k_i}(hg)^{k_i} + \sum_{j=1}^{t} \nu_j f_j, \end{aligned}$$

where $S_1 := \sum_{j=1}^{t} \nu_j f_j$ may bot be S.

c. Compute the shared secret key

$$\begin{aligned} &\sum_{i=1}^{l} \eta_i h^{-k_i}(a + R)(hg)^{k_i} \\ = &\sum_{i=1}^{l} \eta_i [h^{-k_i+m}(hg)^{k_i+m} + h^{k_i} \cdot R \cdot (hg)^{k_i}] \\ = &\ h^m(\sum_{i=1}^{l} \eta_i h^{-k_i}(hg)^{k_i})(hg)^m \\ = &\ h^{-m}(h^{-n}(hg)^n + S_1)(hg)^m \\ = &\ h^{-(m+n)}(hg)^{m+n} = K. \end{aligned}$$

More about public key cryptosystems based on semidirect products, see [20, 21, 29, 32–35, 42, 48] for an example.

6 Logarithmic Signatures

The logarithmic signatures were first used in the cryptography in order to construct a symmetric key cryptosystem PGM [40]. Nearly twenty years later, S.S. Magliveras *et al.* [38, 41] proposed three public key encryption schemes, called MTS_1, MTS_2 and MTS_3, based on logarithmic signatures for finite groups. Their security relies on the following hard factorization problem (Problem 11).

Let G be a finite (nonabelian) group and let $A_i := [a_{i1}, a_{i2}, \cdots, a_{ir_i}]$ be a finite sequence of elements of G, where r_i is called the length of A_i and $\overline{A_i}$ denotes the element $\sum_{j=1}^{r_i} a_{ij}$ in the group ring $\mathbb{Z}G$. An ordered sequence $\alpha := [A_1, A_2, \cdots, A_s]$ of A_i can be viewed as an $s \times r$ matrix $\alpha = (a_{ij})$, where $r = \max\{r_i\}$ and $a_{ij} = 0$ for $j > r_i$. Let $\overline{A_1} \cdot \overline{A_2} \cdots \overline{A_s} = \sum_{a_g \in G} a_g g$, where $a_g \in \mathbb{Z}$.

Definition 6.1. A sequence $\alpha = [A_1, A_2, \cdots, A_s]$ described as above is said to be

1) a cover for G if $a_g > 0$ for all $g \in G$.

2) a logarithmic signature for G if $a_g = 1$ for all $g \in G$.

3) a $[s, r]$-mesh cover if α is a cover for G, all A_i have the same length r and the distribution of the set $\{a_g \mid g \in G\}$ is approximately uniform.

Note that if $\alpha = [A_1, A_2, \cdots, A_s]$ is a logarithmic signature for G then for each element y of G there is a unique factorization with $q_i \in A_i, 1 \le i \le s$

$$y = q_1 \cdot q_2 \cdots q_s. \tag{1}$$

In general, it is not the case for covers.

A logarithmic signature α is called tame if the complexity of the factorization of y in (1) is in polynomial time. Otherwise, α is called wild.

Let $\alpha = [A_1, A_2, \cdots, A_s]$ be a cover of type $(r_1, r_2, \cdots, r_s)$ for G and let $m = \prod_{i=1}^{s} r_i$. Then the cover α induces an efficiently computable surjective mapping

$$\breve{\alpha} : \mathbb{Z}_m \longrightarrow G. \tag{2}$$

If α is a logarithmic signature, then the induced mapping $\breve{\alpha}$ is bijective. Moreover, if a logarithmic signature α is tame, then the inverse $\breve{\alpha}^{-1}$ is efficiently computable.

Proposition 6.2. If α is a wild logarithmic signature and β is a tame logarithmic signature for a finite group G, then the mapping $\breve{\alpha}\breve{\beta}^{-1} : \mathbb{Z}_{|G|} \longrightarrow \mathbb{Z}_{|G|}$ is a one-way permutation.

Let $\gamma : \{e\} = G_0 < G_1 < \cdots < G_{s-1} < G_s$ be a sequence of subgroups of G and let A_i be an ordered, complete set of right coset representatives of G_{i-1} in G_i. Then the sequence $[A_1, A_2, \cdots, A_s]$ forms a logarithmic signature α for G, and is called exact-transversal with respect to γ. If we set $B_i := g_{i-1}^{-1} A_i g_i$, $i = 1, \cdots, s$, where $g_0, g_1, \cdots, g_s \in G$, then the sequence $\beta : [B_1, B_2, \cdots, B_s]$ is again a logarithmic signature for G. When $g_0 = g_s = 1$, then β is said to be a sandwich of α.

Definition 6.3. A logarithmic signature α for a finite group G is called

1) transversal, if α is the sandwich of an exact-transversal logarithmic signature for G.

2) non-transversal, if it is not transversal.

3) totally non-transversal, if none of its blocks is a coset of a non-trivial subgroup of G.

Problem 11. *(Factorization Problem for Logarithmic Signatures/Covers) Given* $g \in G$ *and* $\alpha = [A_1, A_2, \cdots, A_s] = (a_{ij})$*. Find* $a_{ij_i} \in A_i, i = 1, \cdots, s$*, such that* $g = a_{1j_1} a_{2j_2} \cdots a_{sj_s}$.

6.1 MST$_1$- Cryptosystem

Let G be a finite permutation group with a sequence γ of subgroups of G.

1) Key Generation: Bob generates

   a. a totally non-transversal logarithmic signature α, a transversal logarithmic signature β for G and .

   b. a short sequence of exact-transversal logarithmic signatures $\theta_1, \theta_2, \cdots, \theta_k$ such that $\sigma := \breve{\alpha}\breve{\beta}^{-1} = \breve{\theta}_1 \cdot \breve{\theta}_2 \cdots \breve{\theta}_k$, where k is a small integer > 1.

   c. The public key: α, β, G.
   The private key: $\theta_1, \cdots, \theta_k$.

2) Encryption: Let $m \in \mathbb{Z}_{|G|}$ be a message. Alice computes the cipher

$$C = \sigma(m) = (\breve{\alpha}\breve{\beta}^{-1})(m).$$

3) Decryption: Bob recovers the message m by computing

$$\breve{\theta}_k^{-1}(\breve{\theta}_{k-1}^{-1}(\cdots(\breve{\theta}_1^{-1}(C)\cdots))) = m.$$

4) Security Analysis:
The security of MST$_1$ relies on the hardness of the factorization problem with respect to wild logarithmic signatures. In [41] the authors assume that totally non-transversal logarithmic signatures are "wild like". Unfortunately, J.M. Bohli *et al.* [14] have proved that totally non-transversal logarithmic signatures can be tame. This means that not any totally non-transversal logarithmic signature is suitable for being used as a key in MST$_1$. In addition to that there are still no practical implementations of MST$_1$ in sight.

6.2 MST$_2$- Cryptosystem

Let G and H be large groups.

1) Key Generation: Bob

   a. generates an epimorphism $f : G \to H$ and a random $[s, r]$-mesh cover $\alpha = (a_{ij})$ for G

   b. computes $\beta = (b_{ij}) = f(\alpha) = (f(a_{ij}))$.

   c. The public key: α, β.
   The private key: f.

2) Encryption: Let $m \in H$ be a message. Alice

   a. chooses $R \in \mathbb{Z}_{r^s}$.

   b. computes $y_1 = \breve{\alpha}(R)$, $y_2 = \breve{\beta}(R)$, $y_3 = my_2$.

   c. Alice transmits the cipher (y_1, y_3) to Bob.

3) Decryption: Bob

   a. computes $f(y_1) = y_2$, and

   b. computes $m = y_3 y_2^{-1}$.

4) Security Analysis:
If Eve wants to break the system, she has to find the value y_2. There are also two theoretical methods to recover y_2. One is to find a value $R^* \in \mathbb{Z}_{r^s}$ such that $\breve{\alpha}(R^*) = y_1$. If it is the case, then we can compute $\breve{\beta}(R^*) = y_2$. In order to effectively compute R^* such that $y_1 = \breve{\alpha}(R^*)$, Eve has to factorize the public data y_1 with respect to α. This means that Eve has to solve the Factorization Problem 11. This problem is in general an intractable problem for large groups. Second, Eve can try to find a homomorphism $f^* : G \to H$ such that $\beta = f^*(\alpha)$. S.S. Magliveras *et al.* [41] claimed that if the symmetric group S_n is used as the platform group of the scheme MST$_2$ and the private key $f : G \to G$ is conjugation by an element g in S_n, then the scheme MST$_2$ is vulnerable to the second attack.

6.3 MST$_3$- Cryptosystem

Let G be a nonabelian group with nontrivial center $\mathcal{Z}(G)$.

1) Key Generation: Bob

   a. generates a tame logarithmic signature $\beta = [B_1, B_2, \cdots, B_s] := (b_{ij})$ of type $(r_1, r_2, \cdots, r_s)$ for $\mathcal{Z}(G)$ and a random cover $\alpha = [A_1, A_2, \cdots, A_s] := (a_{ij})$ of the same type as β for a certain subset $\mathcal{F}$ of G such that $A_1, A_2, \cdots, A_s \subseteq G \setminus \mathcal{Z}(G)$.

   b. chooses $t_0, t_1, \cdots, t_s \in G \setminus \mathcal{Z}(G)$.

   c. computes $\breve{\alpha} := [\tilde{A}_1, \tilde{A}_2, \cdots, \tilde{A}_s]$, where $\tilde{A}_i = t_{i-1}^{-1} A_i t_i$.

   d. computes $\breve{\gamma} = (h_{ij})$, where $h_{ij} := b_{ij}\tilde{a}_{ij}$.

   e. The public key: $\breve{\alpha}, \breve{\gamma}$.
   The private key: $\beta, t_0, t_1, \cdots, t_s$.

2) Encryption: Let $m \in \mathbb{Z}_{|\mathcal{Z}(G)|}$ be a message. Alice

   a. computes $y_1 := \breve{\alpha}(m)$ and $y_2 := \breve{\gamma}(m)$.

   b. Alice transmits the cipher (y_1, y_2) to Bob.

3) Decryption: Bob

   a. computes $y_2 t_s^{-1} y_1^{-1} t_0 = \breve{\beta}(m)$.

   b. computes $m = \breve{\beta}^{-1}(y_2 t_s^{-1} y_1^{-1} t_0)$.

4) Security Analysis:
If Eve tries to obtain the private logarithmic signature β and the pair (t_0, t_s) of elements of G from the public information $(\breve{\alpha}, \breve{\gamma})$, it is sufficient for Eve to determine a sandwich transform β‘ of β which is equivalent to β, i.e., $\breve{\beta}$‘ $= \breve{\beta}$. Thus, it is sufficient to assume that the first element b_{1j} of each block B_j, except for the last block B_s of B, is the identity $1 \in G$. From the equation $h_{11} = t_0^{-1} a_{11} t_1$, Eve chooses an element of $G \setminus \mathcal{Z}(G)$ as the value of t_0. On the other hand, due to the elements t_i and $t_i z$ for $z \in \mathcal{Z}(G)$ lie in the same coset $t_i \mathcal{Z}(G)$. It is then sufficient to choose one t_0 only from each distinct coset of G modulo $\mathcal{Z}(G)$.

5) Suggested Platforms: The authors [38] of MTS$_3$ employ the Suzuki-2-group of the order q^2 as the platform group of MTS$_3$, where $q = 2^m$ is the order of the center $\mathcal{Z}(G)$ and the integer m is not a power of 2. Thus, if the Suzuki-2-group is implemented as the platform group of MTS$_3$, then there are $(q-1)q$ possible choices for the pair (t_0, b_{s1}). If q is large, the type of the attack is in not feasible.

More about public key cryptosystems based on logarithmic signatures see [14, 38, 40, 41, 58, 60] for an example.

7 Algebraic Eraser

The Algebraic Eraser is a binary operation consists of a semidirect product and a homomorphism of monoids and an action of a nonabelian group on a monoid. The main purpose of building the Algebraic Eraser is to design lightweight public key cryptosystems. The Algebraic Eraser key agreement scheme was introduced by Anshel, Anshel, Goldfeld and Lemieaux in 2004; the corresponding paper [1] appeared in 2006. The Algebraic Eraser and the Algebraic Eraser-based protocol are specially designed for commercial purposes. The company SecureRF owns the trademark of them. It claims a security level of 2^{128} for their preferred parameter sizes. The authors in the paper [1] gave a concrete realization of the Algebraic Eraser key agreement protocol using infinite braid groups named the colored Burau key agreement protocol (CBKAP). This Diffie-Hellman-like protocol has been proposed as a standard in ISO JTC-1/SC-31 (29167-20) to protect various communication protocols like RFID, NFC, or Bluetooth for devices associated with ISO-18000 and the Internet of Things [3, 8].

Let M, N be monoids and let S be a nonabelian group which acts on M on the left, and does not act on N. The semidirect product $M \rtimes S$ of M and S is then a monoid whose internal binary operation is given by

$$(m_1, s_1) \cdot (m_2, s_2) := (m_1^{s_1} m_2, s_1 s_2).$$

The Algebraic Eraser (AE) $\mathbf{E}$ is the binary operation specified within the 6-tuple

$$(M \rtimes S, N, \Pi, \mathbf{E}, A, B)$$

termed the $\mathbf{E}$-structure. where $\Pi : M \longrightarrow N$ is a monoid homomorphism, $\mathbf{E}$ is the function

$$\mathbf{E} : (N \times S) \times (M \rtimes S) \longrightarrow N \times S$$

given by $\mathbf{E}((n, s), (m_1, s_1)) := (n\Pi({}^{s}m_1), s s_1)$, and A, B are submonoids of $M \rtimes S$ satisfying $\mathbf{E}$-Commuting, i.e., they satisfy the equation

$$\mathbf{E}(\Pi(a), s_a, (b, s_b)) = \mathbf{E}(\Pi(b), s_b, (a, s_a)),$$

for all $(a.s_a) \in A$, $(b, s_b) \in B$. For simplicity, the symbol $\star$ will be used to replace the symbol $\mathbf{E}$ as follows:

$$\mathbf{E}((n, s), (m_1, s_1)) = (n, s) \star (m_1, s_1).$$

The operation $\star$ satisfies the associated property as follows: Given $(n, s) \in N \times S$ and $(m_1, s_1), (m_2, s_2) \in M \rtimes S$,

$$((n, s) \star (m_1, s_1)) \star (m_2, s_2) = (n, s) \star ((m_1, s_1) \cdot (m_2, s_2)).$$

7.1 The Key Agreement Scheme based on the Algebraic Eraser

The Algebraic Eraser key agreement scheme designed by Anshel, Anshel, Goldfeld and Lemieaux in [3] is a type of Diffie-Hellman protocol: Let N_A and N_B be submonoids of N so that they commute elementwise.

1) Alice selects her private key: $n_a \in N_A$ and $(a_i, s_{a_i}) \in A$.
Bob selects his private key: $n_b \in N_B$ and $(b_j, s_{b_j}) \in B$.

2) Alice computes P_A

$$P_A = (n_a, id) \star (a_i, s_{a_i}),$$

and transmits P_A to Bob.
Bob computes P_B

$$P_B = (n_b, id) \star (b_1, s_{b_1}),$$

and transmits P_B to Alice.

3) The shared secret key K of Alice and Bob is

$$((n_a, id) \cdot P_B) \star (a_i, s_{a_i}) = ((n_b, id) \cdot P_A) \star (b_j, s_{b_j}).$$

7.2 The CBKAP

The braid groups is used in order to implement the CBKAP. E-multiplication is an action of the braid group on pairs of matrices over a field and permutations. Recall that there is a surjective homomorphism from the Artin braid group B_n onto the symmetric group S_n. With help of the colored Burau representation of B_n, that is an extended version of the reduced Burau representation [3, 10, 17], the semi-direct product $M \rtimes S_n$ is defined as a group generated by the set $\{(x_1(t), s_1), \cdots, (x_{n-1}(t), s_{n-1})\}$, where $x_i(t)$ is a colored Burau matrix of B_n and $s_i = (i\; i+1)$ is a transposition of S_n, for all $i = 1, \cdots, n-1$.

The nonabelian group $M \rtimes S_n$ is also called the colored Burau group. The authors of the paper [3] then give an example to concretely realize the above algorithm by using the colored Burau matrices. The protocol is also called the colored Burau key agreement protocol (CBKAP).

Fix an integer $n \geq 7$ and a prime $p > n$. Let $M \leq \mathrm{GL}(n, \mathbb{F}_p(t))$, where $t = (t_1, \cdots, t_n)$, be a subgroup, and let $S = S_n$ be the symmetric group on n symbols and $N = \mathrm{GL}(n, \mathbb{F}_p)$. Fix n elements $\tau_1, \cdots, \tau_n \in \mathbb{F}_p$, and the homomorphism $\Pi : M \longrightarrow N$ is defined by setting $\tau_i = t_i, i = 1, \cdots, n$. Let $z \in M \rtimes S_n$ be a fixed element and let $A = z \cdot \{(x_{l_1}(t), s_{l_1}), \cdots, (x_{l_\mu}(t), s_{l_\mu})\} \cdot z^{-1}$ and $B = z \cdot \{(x_{r_1}(t), s_{r_1}), \cdots, (x_{r_\nu}(t), s_{r_\nu})\} \cdot z^{-1}$, where $| l_i - r_j | \geq 2$ for $1 \geq i, j \leq n$, be two E-commuting subgroups of $M \rtimes S_n$.

1) The public key: $(M \rtimes S, N, \Pi, \mathbf{E}, A, B)$ and a matrix $m_0 \in \mathrm{GL}(n, \mathbb{F}_p)$ of order $p^n - 1$ such that two subgroups N_A, N_B of N consist of linear combinations of powers of m_0 over $\mathbb{F}_p$.
   The private key: z and $(x_{l_1}(t), s_{l_1}), \cdots, (x_{l_\mu}(t), s_{l_\mu})$, $(x_{r_1}(t), s_{r_1}), \cdots, (x_{r_\nu}(t), s_{r_\nu})$.

2) Alice selects her secret key: $n_a \in N_{_A}$, $k \in \mathbb{N}$ and some $(x_{a_i}(t), s_{a_i}) \in A$, $i = 1, \cdots, k$.
   Bob selects his secret key: $n_b \in N_{_B}$, $l \in \mathbb{N}$ and some $(x_{b_j}(t), s_{b_j}) \in B$, $j = 1, \cdots, l$.

3) Alice computes P_A

$$\begin{aligned} P_A &= (\cdots((n_a, id) \star z \star (x_{a_1}(t), s_{a_1}) \star (x_{a_2}(t), s_{a_2})) \star \\ &\qquad \cdots) \star (x_{a_k}(t), s_{a_k}) \star z^{-1} \\ &= (n_a \cdot \Pi(z) \cdot \Pi({}^{s_z}\mathcal{A}) \cdot \Pi({}^{s_z s_\mathcal{A}} z^{-1}),\ s_z s_\mathcal{A} s_{z^{-1}}), \end{aligned}$$

   where $\Pi({}^{s_z}\mathcal{A}) = \Pi({}^{s_z} x_{a_1}(t))\Pi({}^{s_z s_{a_1}} x_{a_2}(t)) \cdots \cdot \Pi({}^{s_z s_{a_1} \cdots s_{a_{k-1}}} x_{a_k}(t))$ and $s_\mathcal{A} = s_{a_1} \cdots s_{a_k}$.
   Alice transmits P_A to Bob.
   Bob computes

$$\begin{aligned} P_B &= (\cdots((n_b, id) \star z \star (x_{b_1}(t), s_{b_1}) \star (x_{b_2}(t), s_{b_2})) \star \\ &\qquad \cdots) \star (x_{b_l}(t), s_{b_l}) \star z^{-1} \\ &= (n_b \cdot \Pi(z) \cdot \Pi({}^{s_z}\mathcal{B}) \cdot \Pi({}^{s_z s_\mathcal{B}} z^{-1}),\ s_z s_\mathcal{B} s_{z^{-1}}), \end{aligned}$$

   and transmits P_B to Alice.

4) The secret shared key K of Alice and Bob is

$$\begin{aligned} K &= (\cdots((n_a, id) \cdot P_B \star z \star (x_{a_1}(t), s_{a_1}) \\ &\quad \star(x_{a_2}(t), s_{a_2})) \star \cdots) \star (x_{a_k}(t), s_{a_k}) \star z^{-1} \\ &= (\cdots((n_b, id) \cdot P_A \star z \star (x_{b_1}(t), s_{b_1}) \\ &\quad \star(x_{b_2}(t), s_{b_2})) \star \cdots) \star (x_{b_l}(t), s_{b_l}) \star z^{-1}. \end{aligned}$$

5) Security Analysis:
   For simplicity, we assume that the matrix $n_a = m_0^\alpha$ for a secret $\alpha \in \mathbb{Z}^+$. If Eve intercepts the public data P_A and P_B and tries to break the shared key K, it is sufficient to merely solve the matrix m_0^α and the element $z \in M \rtimes S_n$. First, Eve diagonalizes the matrix m_0: $Q m_0 Q^{-1} = (\lambda_1, \cdots, \lambda_n)$, where $Q \in N$. Then,

$$m_0^\alpha = Q^{-1}(\lambda_1^\alpha, \cdots, \lambda_n^\alpha)Q.$$

   On the other hand, by the condition that the subgroups N_A and N_B of N commute, it applies $\Pi({}^{s_z}\mathcal{A})\Pi({}^{s_z}\mathcal{B}) = \Pi({}^{s_z}\mathcal{B})\Pi({}^{s_z}\mathcal{A})$. Thus, the matrices $\Pi({}^{s_z}\mathcal{A})$ and $\Pi({}^{s_z}\mathcal{B})$ take the forms $\begin{pmatrix} X & 0 \\ 0 & I \end{pmatrix}$ and $\begin{pmatrix} I & 0 \\ 0 & Y \end{pmatrix}$, respectively.

   Suppose that z were known. Then, Eve can obtain m_0^α and recover the matrices $\Pi({}^{s_z}\mathcal{A})$ and $\Pi({}^{s_z s_\mathcal{B} s_\mathcal{A}} z^{-1})$ in polynomial time. Therefore, it remains to ask how to determine the element z. The security of the CBKAP depends on the simultaneous conjugacy search problem. There are not any successful attacks to solve the simultaneous conjugacy search problem.

   Problem 12. *(Simultaneous Conjugacy Search Problem) Let $w_1 = z^{-1}a_1 z, \cdots, w_k = z^{-1}a_k z$. If only $w_1, \cdots, w_k$ are public, find the conjugating element z.*

For more about Algebraic Eraser key agreement scheme, see [1, 3, 5–8, 13].

8 Conclusion

There are innovative ideas to propose nonabelian group based-public key cryptography, although, most cryptographic systems seem to be vulnerable to security. For example, the conjugacy search problem on linear groups used in the mentioned protocols, *e.g.*, matrix groups and braid groups, seems to be not be hard. Nevertheless, they still have the value of reference. Some of these systems have some modifications that still have a sufficient security level. On the other hand, the efficiency and security of a cryptographic system does not only depend on the design of the algorithm, but also on the choice of platform.

References

[1] I. Anshel, M. Anshel, D. Goldfeld, S. Lemieux, "Key agreement, the algebraic eraser^{TM}, and lightweight cryptography," *Contemporary Mathematics*, vol. 418, pp. 1-34, 2006.

[2] I. Anshel, M. Anshel, D. Goldfeld, "An algebraic method for public-key cryptography," *Mathematics Research Letter*, vol. 6, pp. 287-291, 1999.

[3] I. Anshel, D. Atkins, D. Goldfeld, P. Gunnells, "Defeating the Ben-Zvi, Blackburn, and Tsaban Attack on the Algebraic Eraser," *IACR ePrint* 2016/044.

[4] E. Artin, "Theory of braids," *Annal of Mathematics*, vol. 48, pp. 101-126, 1947.

[5] D. Atkins, *Algebraic Eraser: A Lightweight, Efficient Asymmetric Key Agreement Protocol for use in No-Power, Low-Power, and IoT Devices*, 2015. (csrc.nist.gov/groups/ST/lwc-workshop2015/papers/session8-atkins-paper.pdf)

[6] D. Atkins, D. Goldfeld, *Addressing the Algebraic Eraser Diffie-Hellman Over-the Air Protocol*, 2015. (http://eprint.iacr.org/2016/205.pdf)

[7] D. Atkins, P. E. Gunnells, *Algebraic Eraser: A Lightweight, Efficient Asymmetric Key Agreement Protocol for use in No-Power, Low-Power, and IoT Devices*, 2015. (http://csrc.nist.gov/groups/ST/lwc-workshop2015/presentations/session8-atkins-gunnell.pdf)

[8] A. Ben-Zvi, S. R. Blackburn, B. Tsaban, "A Practical Cryptanalysis of the Algebraic Eraser," 2016. (https://arXiv:1511.03870v2)

[9] G. Baumslag, B. Fine, X. Xu, "Cryptosystems using Linear Groups," *Applicable Algebra in Engineering, Communication and Computing*, vol. 17, no. 3-4, pp. 205-217, 2006.

[10] G. Baumslag, B. Fine, M. Kreuzer, G. Rosenberger, *A Course in Mathematical Cryptography*, De Gruyter, 2015.

[11] S. Bigelow, "Braid groups are linear," *Journal of the American Mathematical*, vol. 14, pp. 471-486, 2001.

[12] S. R. Blackburn, C. Cid, C. Mullan, *Group Theory in Cryptography*, 2010. (https://arxiv.org/pdf/0906.5545)

[13] S. R. Blackburn, M. J. B. Robshaw, "On the security of the algebraic eraser tag authentication protocol," in *14th International Conference on Applied Cryptography and Network Security (ACNS'16)*, Lecture Notes in Computer Science, vol. 9696, pp. 3-17, 2016.

[14] J. M. Bohli, M. I. Gonzaĺez Vasco, C. Martínez, R. Steinwandt, "Weak keys in MST$_1$," *Design, Code and Cryptography*, vol. 37, pp. 509-524, 2005.

[15] Z. Busser, *Braid Group Cryptography*, 2009.

[16] J. Cheon, B. Jun, "A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem," in *Advances in Cryptography (CRYPTO'03)*, Lecture Notes in Computer Science, vol. 2729, pp. 212-224, 2003.

[17] P. Dehornog, "Braid-based cryptography," *American Mathematical Society*, vol. 360, pp. 5-33, 2004.

[18] P. Dehornog, "Using shifted conjugacy in braid-based cryptography," *Contemporary Mathematics*, vol. 418, pp. 65-94, 2006.

[19] W. Diffie, M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory IT*, vol. 22, pp. 644-654, 1976.

[20] J. Ding, A. Miasnikov, A. Ushakov, *A Linear Attack on a Key Exchange Protocol Using Extensions of Matrix Semigroups*, 2015. (http;//eprint.iacr.org/2015/018)

[21] M. Eftekhari, *Cryptanalysis of Some Protocols using Matrices over Group Rings*, 2015. (http://arxiv.org)

[22] B. Fine, M. Habeeb, D. Kahrobaei, G. Rosenberger, *Aspects of Nonabelian Group Based Cryptography: A Survey and Open Problems*, 2011. (https://arXiv:1103.4093v2)

[23] N. Franco, J. González-Meneses, "Conjugacy problem for braid groups and garside groups," *Journal of Algebra*, vol. 266, pp. 112-132, 2003.

[24] D. Freeman, *The Discrete Logarithm Problem in Matrix Groups*, 2004.

[25] D. Grigoriev, "Public-key cryptography and invariant theory," *Journal of Mathematical Sciences*, vol. 126, no. 3, pp. 1152-1157, 2005.

[26] D. Grigoriev, A. Kojevniko, S. Nikolenko, *Invariant-based Cryptosystem and Their Security Against Provable Worst-Case Break*, Technical Report 158, Max-Planck-Inst, Preprints, 2007.

[27] D. Grigoriev, I. Ponomarenke, "Constructions in public-key cryptography over matrix groups," *Contemporary Mathematics: Algebraic Methods in Cryptography*, vol. 418, pp. 103-120, 2007.

[28] S. D. Hasapis, D. Panagopoulos, "A survey of group-based cryptogrsphy," *Journal of Applied Mathematics & Bioinformatics*, vol. 5, no. 3, pp. 73-96, 2015.

[29] M. Habeeb, D. Kahrobaei, C. Koupparis, V. Shpilrain, "Public key exchange using semidirect product of (semi)groups," *Lecture Notes in Computer Science*, vol. 7954, pp. 475-486, 2013.

[30] U. Isik, *Computational Problems in the Braid Group with Applications to Cryptography*, 2005.

[31] D. Kahrobaei, M. Anshel, "Decision and search in non-abelian Cramer-Shoup public key cryptosystem," *Group Complexity Cryptology*, vol. 1, no. 2, pp. 97-115, 2009.

[32] D. Kahrobaei, C. Koupparis, V. Shpilrain, "Public key exchange using matrices over group rings," *Group Complexity Cryptology*, vol. 5, pp. 217-225, 2013.

[33] D. Kahrobaei, H. T. Lam, V. Shpilrain, *Public Key Exchange using Extensions by Endomorphisms and Matrices over a Galois Field, Preprint*, 2014.

[34] D. Kahrobaei, V. Shpilrain, *Using Semidirect Product of (Semi)groups in Public Key Cryptography*, 2016. (https://arXiv:1604.05542v1)

[35] K. H. Ko, D. H. Choi, M. S. Cho, J. W. Lee, *New Signature Scheme using Conjugacy Problem*, 2002. (http://eprint.iacr.org/2002/168)

[36] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. S. Kang, C. Park, "New public key cryptosystem using braid groups," in *Advances in Cryptology (CRYPTO'00)*, pp. 166-184, 2000.

[37] D. Krammer, "Braid groups are linear," *Annals of Mathematics*, vol. 155, pp. 131-156, 2002.

[38] W. Lempken, T. van Trung, S. S. Magliveras, W. Wei, "A public key cryptosystem based on non-abelian finite groups," *Journal of Cryptology*, vol. 22, pp. 62-74, 2009.

[39] K. Mahlburg, *An Overview of Braid Group Cryptography*, Preprint, 2004.

[40] S. S. Magliveras, "A cryptosystem from logarithmic signatures of finite groups," in *Proceeding of the 29th Midwest Symposium on Circuils and Systems, Elsevier Publishing Company*, pp. 972-975, 1986.

[41] S. S. Magliveras, D. R. Stinson, T. van Trung, "New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups," *Journal of Cryptology*, vol. 15, pp. 285-297, 2002.

[42] C. Monico, M. Neusel, "Cryptanalysis of a system using matrices over group rings," *Group Complexity Cryptology*, vol. 7, no. 2, pp. 175-182, 2015.

[43] A. Myasnikov, V. Shpilrain, "A linear decomposition attack," *Group Complexity Cryptology*, vol. 7, pp. 81-94, 2015.

[44] A. Myasnikov, V. Shpilrain, A. Ushakov, "A practical attack on a braid group based cryptographic protocol," in *Advances in Cryptology (CRYPTO'05)*, pp. 86-96, 2005.

[45] A. Myasnikov, V. Shpilrain, A. Ushakov, *Group-based Cryptography*, Birkhäuser Verlag, 2008.

[46] A. D. Myasnikov, A. Ushakov, "Quantum algorithm for the discrete logarithm problem for matrices over finite group rings," *Group Complexity Cryptology*, vol. 6, pp. 31-36, 2014.

[47] S. H. Paeng, K. C. Ha, J. H. Kim, S. Chee, C. Park, "New public key cryptosystem using finite non-abelian groups," in *Advance in Cryptology (CRYPTO'01)*, Lecture Notes in Computer Science, vol. 2139, pp. 470-485, 2001.

[48] V. Roman'kov, *Linear Decomposition Attack on Public Key Exchange Protocols using Semidirect Products of (Semi)Groups*, 2015. (https://arXiv:1501.01152v1)

[49] S. K. Rososhek, "New practical algebraic public-key cryptosystem and Some related algebraic and computational aspects," *Applied Mathematics*, vol. 4, no, 7, pp. 1043-1049, 2013.

[50] S. K. Rososhek, "Modified matrix modular cryptosystems," *Britsh Journal of Mathematics & Computer Science*, vol. 5, no. 5, pp. 613-636, 2015.

[51] P. Svaba, *Covers and Logarithmic Signatures of Finite Groups in Cryptography*, Dissertation, 2011.

[52] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithm problems," *SIAM Journal on Computing*, vol. 26, pp. 1484-1509, 1997.

[53] V. Shpilrain, A. Ushakov, *The Conjugacy Search Problem in Public Key Cryptography: Unnecessary and Insufficient*, 2004. (https://arXiv:math/0411644v1)

[54] V. Shpilrain, "Cryptanalysis of stickel's key exchange scheme," *Lecture Notes in Computer Science*, vol. 5010, pp. 283-288, 2008.

[55] R. Steinwandt, "Loopholes in two public key cryptosystems using the modular group," *Lecture Notes in Computer Science*, vol. 1992, pp. 180-189, 2002.

[56] E. Stickel, *A New Public-Key Cryptosystem in Non-Abelian Groups*, 2003. (https://www.semanticscholar.org)

[57] E. Stickel, "A new method for exchanging secret keys," in *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)*, pp. 426-430, 2005.

[58] P. Svaba, *Covers and Logarithmic Signatures of finite Groups in Cryptography*, Dissertation, 2011.

[59] B. Tsaban, "Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography," *Journal of Cryptology*, vol. 28, pp. 601-622, 2015.

[60] M. I. G. Vasco, D. Hofheinz, C. Martinez, R. Steinwandt, "On the security of two public key cryptosystems using non-abelian groups," *Designs, codes and Cryptography*, vol. 32, pp. 207-216, 2004.

[61] N. R. Wagner, M. R. Magyarik, "A public key cryptosystem based on the word problem," in *Advances in Cryptology (CRYPTO'84)*, Lecture Notes in Computer Science, vol. 196, pp. 19-36, 1985.

[62] L. Wang, L. Wang, Z. Cao, E. Okamoto, J. Shao, "New constructions of public-key encryption schemes from conjugacy search problems," in *Information Security and Cryptology*, Lecture Notes in Computer Science, vol. 6584, pp. 1-17, 2010.

[63] X. Wang, C. Xu, G. Li, H. Lin, W. Wang, "Double shielded public key cryptosystems," *Cryptology ePrint Archive Report 2014/588*, pp. 1-14, 2014.

[64] A. Yamamura, "Public-key cryptosystems using the modular group," in *1st Internationa Workshop on Practice and Theory in Public Key Cryptography (PKC'98)*, Lecture Notes in Computer Science, vol. 1431, pp. 203-216, 1998.

Biography

Tzu-Chun Lin received the PhD in Mathematics from the Faculties for Mathematics and Science of the Georg-August-University at Göttingen in Germany. She is an associate professor in the Department of Applied Mathematics of Feng Chia University in Taiwan, R.O.C.. Her current research interests include commutative algebras, invariant theory of finite groups and public-key cryptography.

Evidence Based Trust Estimation Model for Cloud Computing Services

Shilpa Deshpande and Rajesh Ingle

(Corresponding author: Shilpa Deshpande)

Department of Computer Engineering, College of Engineering Pune, Savitribai Phule Pune University
Wellesely Road, Shivajinagar, Pune 411005, Maharashtra, India
(Email: shilpshree@yahoo.com)

(Received Nov. 8, 2016; revised and accepted Feb. 1, 2017)

Abstract

With growing acceptance of cloud environment, demand for cloud services based applications is rapidly increasing. Cloud environment is inherently distributed and highly dynamic. Usually, many cloud services having similar functionalities but varying performances are offered. This makes difficult for user to identify the appropriate cloud service. Consequently, trust assessment is becoming essential to locate and to continue using the suitable cloud service. Quality of Service (QoS) attributes are significant for trust assessment in cloud environment. These attributes are dynamically changing and are not considered by the traditional trust assessment approaches. Hence, these approaches are inadequate to withstand in cloud environment. This paper proposes an evidence based trust estimation model (EBTEM) for trust assessment of cloud services. EBTEM uses various attributes of cloud service as evidence factors. It performs adaptive trust computation, which is sensitive to changes in the service behavior. EBTEM also presents a method for dynamic trust prediction. Experimental results demonstrate that the proposed model outperforms other models in respect of accuracy and efficiency.

Keywords: Adaptive Trust Computation; Cloud Computing; Dynamic Trust Assessment; Evidence Based Trust; Quality of Service (QoS)

1 Introduction

Real world software systems are increasingly becoming large, complex and business critical. Such enterprise applications demand flexibility in terms of compute capability, location of data, resources and users. Cloud computing paradigm fulfills these needs on demand by dynamically provisioning services and resources over the network. Cloud computing infrastructure and services offer several benefits such as simplicity to the end- users, reduced costs, dynamic resource sharing, pay-per-use and dynamic resource availability. On one side while cloud environment offers these benefits to the user community, on the other side it also poses challenges of increased system complexity, dynamicity, non-transparency of cloud services and geographically distributed data centers [2, 24].

On this background, ensuring availability of services and predicting performances of applications hosted on cloud infrastructure become more and more challenging. Security of applications and data deployed on cloud and maintaining privacy of users, add up further to these challenges. Thus, from the consumer perspective, service not being available when needed, longer time to get response than expected and security and privacy risks, result in lack of trust toward the provider [7].

Across a broad spectrum, enterprises such as banking, hospitals and the like, adopting the cloud computing for its cost-benefits, need to maintain the confidentiality and integrity [9] of the huge amount of data placed in the cloud environment. But as the cloud services are designed to be offered in non-transparent fashion, enterprises may believe that they can no longer manage their data. Hence the users may be hesitant about the probable service quality [6, 20]. This motivates the need of establishing efficient mechanism for trust estimation in the cloud environment. However, trust assessment in cloud environment poses the important issues, which are revealed as part of the following discussion.

A service level agreement (SLA) formed between a cloud user and a provider contains the technical and functional details of the offered service [8]. Contents of SLA are not consistent among the cloud service providers offering similar services. Hence users cannot assess the trust of cloud service provider based on its SLA only [8, 29]. The conventional reputation based trust mechanism reveals only the general thinking of cloud service consumers towards the cloud service and does not reflect the judgement about the performance of the cloud service [10]. In the context of cloud computing, the trust mainly depends on the performance of the cloud service depicted in terms of various cloud service attributes [6, 10]. Hence the cloud Quality of Service (QoS) attributes like availability, performance, security are critical and their evidences are needed to be considered by the trust estimation mechanism.

Trust in cloud computing can be viewed as an indicator of service behavior. Hence trust value of a cloud service may change dynamically in response to the experience of the cloud service by the end user [10, 25]. Consequently, the trust estimation needs to be a continuous dynamic process and not a one-time assessment. Cloud auditor as a third party may perform the trust assessment of cloud services. However, as the audit is conducted only after a certain period, corresponding trust assessment does not represent a dynamic trust evaluation of a cloud service [6].

Cloud environment being highly dynamic, method of trust assessment needs to be responsive to the changes in the behavior of cloud service. Thus in turn, it requires an adaptive trust assessment [24]. Consequently, trust assessment of a cloud service needs to consider the relative importance of each individual QoS attribute in trust calculation. Assigning weights manually to the various attributes of a cloud service requires a judgement by an expert and is time-intensive [27]. Moreover, trust assessment based on manually weighing of attributes does not indicate an adaptability to the cloud service in operation.

In this paper, we present an evidence based trust estimation model (EBTEM), addressing the above mentioned issues. More specifically, the contributions are towards development of:

1) A methodology for computation of trust at an instant of time using evidences of multiple attributes of cloud service.

2) Adaptive trust assessment mechanism containing mathematical formulation of weights which are computed adaptively in response to the changes in the behavior of cloud service.

3) Formulation of dynamic trust prediction over a period of time.

4) An algorithm for adaptive and dynamic trust assessments of a cloud service based on service evidence factors.

5) Comparison of the proposed trust model with other models with regard to accuracy and efficiency.

The paper is organized as follows. Section 2 presents a review of related work. In Section 3, the architecture of the system meant for the proposed trust model and the functional overview of trust estimation are described. Section 4 defines the EBTEM and presents the details of adaptive and dynamic trust assessment. Section 5 depicts the algorithm for trust estimation of a cloud service based on service evidence factors. Section 6 covers the performance evaluation of the proposed trust model including the results and analysis. Section 7 concludes the paper.

2 Related Work

The initial approaches of trust assessment in cloud environment are exclusively based on the traditional technique of reputation. This technique employs feedbacks about a particular cloud service from many service consumers to obtain the trust of that service. In reputation-based technique, source of feedback is not known to the cloud service users. Hence credibility of feedbacks is a major issue in the trust assessment [24]. Reputation based mechanism is helpful only in initial judgement about the cloud service. The mechanism may be inadequate as trust placed on the service evolves with experience. Trust assessment approaches proposed by [1, 22, 23] are based on reputation. These approaches lack in the capability to perform dynamic assessment of trust. A framework proposed by Noor and Sheng [22, 23] offers Trust as a Service (TaaS) for evaluation of cloud services. The framework provides a credibility model that differentiates the reliable feedbacks from the deceptive ones. Trust mechanism suggested by Abawajy [1] uses fading factor to keep track of the drop in satisfaction ratings over a period of time.

Along with the user feedbacks as part of reputation mechanism, few of the approaches employ other factors such as declarations by provider, user's own ratings and certificates for the trust assessment. However, authenticity of the factors used for trust assessment is a major concern in these approaches. Habib *et al.* [8] proposed an architecture to evaluate trust of cloud service providers using the combination of multiple factors such as provider statements, user feedbacks, certificates and expert assessment. Trust model proposed by Pawar *et al.* [26] takes into account the fulfillment of service level agreement (SLA) parameters. The approaches [8, 26] perform the trust evaluation in the form of opinions. These approaches do not consider dynamic trust assessment over a period of time. Ghosh *et al.* [5] proposed a framework to evaluate the risk of interaction with cloud service provider. The approach in turn involves evaluating the trust of cloud provider. The trust is estimated using the combination of direct and indirect interactions between user and service provider. The approach calculates a time window based trust using customer's ratings about previous interactions. However, the approach does not reflect the periodic trust update during the interaction. Ratings submitted by customers may be subjective in nature.

A model is suggested by Moyano *et al.* [21] for the trust assessment of cloud provider based on the factors such as SLA, transparency, accounting and auditing. The method uses a trust interval formed by the combination of value of the factor and its associated confidence value. Although the model offers simplicity in trust assessment, trust values are assigned using only the self-assessment based information available on the web sites of cloud providers. Moreover, subjective quantification of the factors may take place during the trust evaluation. Li *et al.* [16] proposed a model to judge the credibility of a cloud service by assessment of its trust. It uses multiple factors which include user ratings, record of service call, service certification and service quality monitoring for evaluation of trust. However, the details of service attributes are not specified explicitly. The weights assigned to the various factors are decided subjectively by the users themselves.

Few of the mechanisms consider QoS attributes for trust assessment. The approach proposed by Manuel *et al.* [19] computes the trust of a cloud resource as a simple summation of values assigned to user feedbacks, security level and reputation. A model is suggested by Manuel *et al.* [18] to compute the reputation based trust of a resource. Trust value of a resource is obtained as a combination of its identity, capability and behavior values. The approaches [18, 19] do not consider the dynamic trust evaluation over a period of time. Fan *et al.* [4] suggested a mechanism for evaluating trust of a cloud service using multiple attributes. The mechanism obtains the trust value by the user's direct interaction with the service. This trust value is combined with the reputation value of a cloud service to obtain the final evaluation. Both the assessment values rely on the feedbacks given by the users. However, authenticity of feedbacks is not addressed by the authors. A fuzzy trust evaluation approach for cloud services is suggested by Huo *et al.* [11]. The approach uses a set of cloud service attributes to evaluate a reputation based trust value. Weights to the various factors in the approaches [4, 11, 18] are assigned manually. Hence these weights may be static and subjective.

The existing QoS based approaches make use of performance, security, availability and reliability as the general attributes of cloud service for trust evaluation. Response time, throughput, capability and network bandwidth are the commonly used performance related factors for trust assessment. System proposed by Qu and Buyya [27] estimates trust of a cloud service by taking into consideration the performance variations of the service due to the dynamic attributes. The authors focus on evaluating the trust of a service prior to the user interaction by retrieving the past data of service attributes. However, updating the trust value of a service during the period of user interaction is not considered by the approach. A framework is proposed by Sidhu and Singh [28] for trust evaluation of cloud service providers based on QoS attributes. The approach monitors the QoS attributes and evaluates the compliance with regard to the SLA. However, the approach does not consider the dynamic trust evaluation and updating trust over a period of time.

Manuel [17] proposed a model which computes the trust of a resource as a combination of its past credentials and present capabilities. Past credentials of the resource are represented in the form of QoS attributes. However, the various attributes for trust assessment are merged by assigning static weights to them. Li *et al.* [15] proposed a dynamic trust management method for the resources in cloud environment. The model evaluates the trust degree of a resource based on the data obtained by monitoring of multiple attributes. The method assigns information entropy based weights to various factors and combines them to generate the trust value. However, the operations involved in the computation of trust are considerably complex in nature. The static factors such as capacity of a resource are treated similarly as dynamic factors during the trust computation.

In summary, the above review of the related work indicates that only few of the approaches [15, 17, 27, 28] employ monitoring based cloud QoS attributes for trust assessment. However, cloud QoS attributes are the vital factors for trust estimation. Values of QoS attributes obtained through monitoring are objective in nature and are more reliable factors for trust assessment. Dynamic cloud environment entails the trust to be evaluated and updated continuously with time. However, the approaches [27, 28] do not consider dynamic trust update of a cloud service according to the periodically changing values of the service attributes. Moreover, the approach [17] combines the various attributes for trust assessment by assigning static weights to them. Static weights may be subjective in nature and the corresponding trust computation does not reflect the adaptability to the changing behavior of a cloud service. Our trust model EBTEM, intends to address these shortcomings in the earlier work. EBTEM performs adaptive and dynamic trust assessments of a cloud service by taking into account multiple quality attributes of the service. Evidence based trust computation used in our model, enables dynamic update of trust values by collecting evidences at different times. EBTEM facilitates adaptive computation of weights for the various service attributes by considering the correlation among the attributes.

3 Architecture of Trust Estimation System

Figure 1 shows the overall layout of the system meant for the proposed trust model. It depicts the main trust estimator module which is connected with the other supplementary modules. Here, the cloud user can be the end-user who intends to use the trustworthy cloud service or the cloud user can be the service provider willing to deploy the application onto the cloud.

The service specification collector compiles the functional requirements of the cloud service, submitted by cloud user. Based on the kind of application to be executed, the user decides the functional specifications of the service. Service extraction module then finds the services from service repository whose functional specifications match with the required one.

Trust estimator module is the core component performing an adaptive and dynamic trust assessment of the cloud service. Direct interaction between a cloud user and the service, is the main source of evidence for trust estimation. Consequently, for the cloud service in execution, at each instant of time, trust estimator obtains the evidence factors compiled by an evidence collector over the designated period of time. The module makes use of these evidence

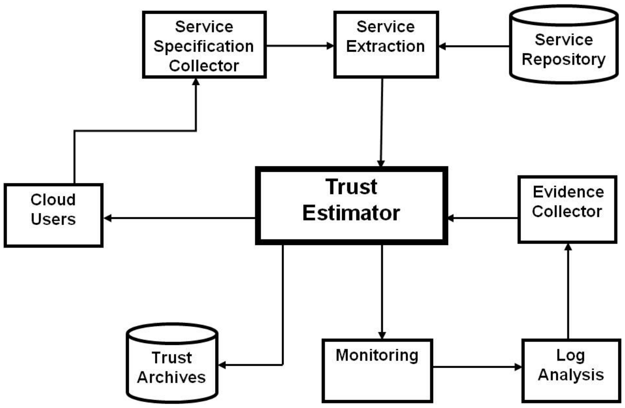


Figure 1: Architecture of trust estimation system

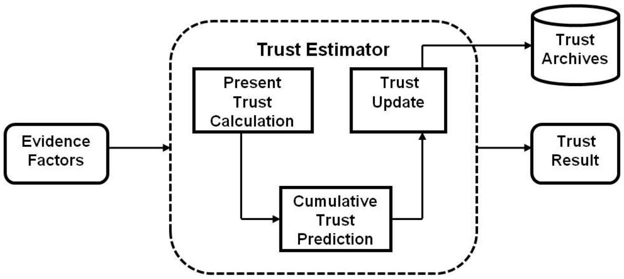


Figure 2: Functional overview of trust estimation

factors of the service attributes, for the trust assessment of the service. The cloud user can decide whether to continue using the cloud service based on the state of trust indicated by the trust estimator. The result of trust assessment is recorded in the trust archives.

Monitoring is a real-time dynamic process tracking the performance of the cloud service in operation. It observes the variation in the performance and creates a log containing values of cloud service attributes such as response time, throughput, availability and security. Log analysis module retrieves the evidence factors recorded as part of continuous monitoring process after every fixed time interval. The evidence collector collects these evidence factors which are then used for trust assessment of a cloud service.

Trust estimation is the primary focus of this paper. Therefore, the details of supplementary modules which cover service extraction, monitoring and related functionalities, are not discussed further, in this paper. We assume these as the already existing valid services and are available in the form of external interfaces to the trust estimator.

Figure 2 shows the high-level functional overview for the trust estimation of the cloud service in operation. Evidence factors over the period of time, representing the QoS attributes of the cloud service, are taken as input by the trust estimator. The trust estimator calculates present trust of a service by aggregating all the evidence factors at an instant of time. Subsequently, the module performs computation of cumulative trust over a period of time. The trust result generated in the form of cumulative trust indicates the predicted trust level of a cloud service. The trust estimator updates the trust value stored in the trust archives by the latest cumulative trust value. The details of present trust and cumulative trust assessments of a cloud service are described in Section 4. The steps depicting the control flow for trust estimation are presented in Section 5 in the form of algorithm.

4 Evidence Based Trust Estimation Model

Trust value of a cloud service is a function of cloud service attributes. Value of a cloud service attribute is termed as an evidence factor.

Definition 1. *Evidence Based Trust Estimation Model (EBTEM) is defined by a 9-tuple* $(L, AC, TI, C, M, PT, CT, E, D)$ *where*

L: Set of v cloud services: $\{s_1, s_2, ..., s_v\}$

AC: Set of m cloud service attributes: $\{R_1, R_2, ..., R_m\}$

TI: Ordered discrete set of n time instances: $\{1, 2, ..., n\}$

C: An evidence matrix which depicts m evidence factors at each of the n time instances.

M: Normalized evidence matrix.

PT: Present Trust of a cloud service at a particular time instant.

CT: Cumulative Trust of a cloud service over a period of time.

E: A set of core trust estimation functions: $\{f_{PT}, f_{CT}\}$*; where* f_{PT} *indicates a function to compute Present Trust (PT) and* f_{CT} *is a function to assess Cumulative Trust (CT).*

D: A set of allied functions: $\{f_{NE}, f_{CW}\}$*; where* f_{NE} *is a function to normalize evidence factors and* f_{CW} *indicates a function to compute weights of cloud service attributes.*

While a cloud service is running, evidence factors are retrieved after every fixed time interval. Representation of the evidence factors is devised in the form of an evidence matrix as shown below.

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{bmatrix} \quad (1)$$

In Equation (1), at a time instant i such that $1 \leq i \leq n$, a row in the matrix indicates a sample of evidence factors as $\{c_{i1}, c_{i2}, ..., c_{im}\}$ and each value c_{ij} in the sample, denotes a value of an attribute R_j. Thus, there are n samples of evidence factors. Column position in the matrix specifies a particular attribute within the sample.

In order to transform values of all the attributes to uniform range and to make them independent of units, values of the attributes in the evidence matrix need to be normalized. Normalization involves scaling of the values. Thus, for further processing of trust assessment, each value in the evidence matrix is normalized in the range denoted by $[R^{new\_min}, R^{new\_max}]$. From the perspective of desired performance of a cloud service, attributes can be categorized in two types: one where higher value of an evidence factor c_{ij} is desired and the other where lower value of c_{ij} is desired. The category where higher value of c_{ij} is desired, the corresponding normalized value h_{ij} is formulated as shown below.

$$h_{ij} = \frac{(c_{ij} - R_j^{min})(R^{new\_max} - R^{new\_min})}{(R_j^{max} - R_j^{min})} + R^{new\_min} \tag{2}$$

The other category where lower value of c_{ij} is desired, the corresponding normalized value h_{ij} is devised as:

$$h_{ij} = \frac{(R_j^{max} - c_{ij})(R^{new\_max} - R^{new\_min})}{(R_j^{max} - R_j^{min})} + R^{new\_min} \tag{3}$$

In Equations (2) and (3), R_j^{min} is the minimum value of the attribute R_j in a time window of n samples and R_j^{max} is the maximum value of R_j in the same time window of n samples. Higher the resultant value h_{ij}, better is its contribution to the high quality of a cloud service. The normalized evidence matrix is:

$$M = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & h_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \dots & h_{nm} \end{bmatrix} \tag{4}$$

4.1 Attributes-aggregation Based Calculation of Present Trust

For effective trust assessment of a cloud service, corresponding evidence factors need to be evaluated on the basis of their utility and inter-relationship. Accordingly, evidence factors can be combined to find a trust value. A method is developed for trust estimation of a cloud service using the values of attributes, at a particular instant of time. Within a time window of size n, after every fixed time interval, evidence factors are retrieved and a trust value is calculated at a particular time instant.

Definition 2. *Trust value of a cloud service (s_l), at a time instant i, termed as Present Trust (PT) is devised as an aggregation of corresponding all m evidence factors. It is given as:*

$$PT^i(s_l) = \sum\nolimits_{j=1}^{m} w_j h_{ij} \tag{5}$$

where h_{ij} is a normalized evidence factor of attribute R_j at time instant i, w_j is a weight assigned to it such that $0 < w_j < 1$ and $\sum_{j=1}^{m} w_j = 1$.

Subjectively or manually assigned weights are static in nature. Trust assessment techniques based on such weights are not suitable for effective assessment of trust in cloud environment. Thus, weights need to adapt to changes in cloud service behavior [24] and hence computation of weights is crucial for adaptive trust estimation.

4.2 Computation of Weights

Weight designated to an attribute highlights the significance of the attribute in trust calculation. Weight of an attribute is computed based on values of that attribute at changing time instances. This results in adaptive weight assignment which is sensitive to the changes in the values of the attribute over a time period while the cloud service is running. This leads to an adaptive trust assessment. Degree of variation of each attribute within a time window is estimated for deciding weight of that attribute.

Definition 3. *Variation factor of an attribute R_j is formulated as given below.*

$$V(R_j) = \sum_{i=1}^{n} (h_{ij} - A(R_j))^2 \tag{6}$$

where h_{ij} is a normalized evidence factor of attribute R_j at time instant i.

$A(R_j)$ is the average of n evidence factors of attribute R_j in a time window containing n samples, shown as:

$$A(R_j) = (\sum_{i=1}^{n} h_{ij})/n \tag{7}$$

Effect of $V(R_j)$ on weight calculation is defined in terms of impact of variation factor (F_j) as shown below.

Definition 4. *Impact of variation factor (F_j) for attribute R_j is devised as:*

$$F_j = 1/(V(R_j) + (1/n)) \tag{8}$$

where $(1/n)$ is a negligible positive term which ensures a finite value for (F_j), especially in the situation when $V(R_j)$ is zero. However, such situation is rare in practice.

From Equation (8), less is the variation factor $V(R_j)$ of the attribute, higher is its impact (F_j) on the weight of the attribute R_j.
Weight w_j of an attribute R_j is computed as given below.

$$w_j = F_j / \sum_{k=1}^{m} F_k \tag{9}$$

where $0 < w_j < 1$ and $\sum_{j=1}^{m} w_j = 1$. Thus, weights computed using Equation (9) when substituted in Equation (5), result in adaptive trust estimation. Less is the variation factor of the attribute, higher is its resultant weight. This implies an effective trust assessment of a cloud service over a long duration from the perspective of a user. Weight computation of the attribute is also dependent on the variation factors of all other attributes. Weight of an attribute is higher if combined effect of variation factors of all other attributes is higher.

Theorem 1. *Weight assigned to any attribute R_j of a cloud service, is directly proportional to the combined effect of variation factors of all $(m-1)$ attributes other than R_j of the cloud service.*

Proof. Substituting F_j and F_k from Equation (8) in Equation (9), weight of any attribute R_j is:

$$w_j = \frac{[1/(V(R_j) + (1/n))]}{\sum_{k=1}^{m}[1/(V(R_k) + (1/n))]} \tag{10}$$

where $1 \leq j \leq m$. Substituting $V(R_j)$ and $V(R_k)$ from Equation (6) in Equation (10),

$$w_j = \frac{[1/(\sum_{i=1}^{n}(h_{ij} - A(R_j))^2 + (1/n))]}{\sum_{k=1}^{m}[1/(\sum_{i=1}^{n}(h_{ik} - A(R_k))^2 + (1/n))]} \tag{11}$$

Upon simplification, w_j becomes:

$$w_j = \frac{\prod_p [V(R_p) + (1/n)]}{[term1] + [term2] + ... + [termm]} \tag{12}$$

where $1 \leq p \leq m$, $p \neq j$ and

$$term1 = (V(R_2) + \tfrac{1}{n})(V(R_3) + \tfrac{1}{n})...(V(R_m) + \tfrac{1}{n})$$

$$term2 = (V(R_1) + \tfrac{1}{n})(V(R_3) + \tfrac{1}{n})...(V(R_m) + \tfrac{1}{n})$$

$$termm = (V(R_1) + \tfrac{1}{n})(V(R_2) + \tfrac{1}{n})...(V(R_{m-1}) + \tfrac{1}{n}).$$

In Equation (12), for any weight w_j, where $1 \leq j \leq m$, the denominator on the right-hand side, is the same. Hence, for any attribute R_j weight w_j is directly proportional to the product of $(m-1)$ terms in the numerator where each term indicates variation factor of corresponding attribute other than R_j. This proves the theorem. □

Thus, as indicated by Theorem 1, computation of weight for a cloud service attribute takes into consideration correlation of the attribute with all other attributes of the service. This achieves balancing effect while weighing the service attributes and subsequently aggregating them to compute a present trust of the cloud service.

4.3 Prediction of Cumulative Trust

A set of present trust (PT) values of a cloud service computed at different time instances forms a time series. From Equation (5), at time instant n, time series (TS) is:

$$TS = \{PT^1(s_l), PT^2(s_l), ..., PT^n(s_l)\} \tag{13}$$

This set serves as a basis to predict the future value of trust termed as cumulative trust. Prediction of cumulative trust is essential to assess the future quality of a cloud service for the duration of its execution. Hence, a method is developed for dynamic prediction of cumulative trust over a period of time.

Definition 5. *Cumulative Trust (CT) of a cloud service (s_l) over a period of time, predicted at a time instant n is defined as:*

$$CT^n(s_l) = \alpha PT^n(s_l) + (1-\alpha)CT^{n-1}(s_l) \tag{14}$$

where $PT^n(s_l)$ is PT of a cloud service (s_l) at n^{th} time instant as defined by Equation (5) and $CT^{n-1}(s_l)$ is a cumulative trust of a cloud service (s_l) at time instant $(n-1)$. $CT^{n-1}(s_l)$ is subsequently substituted repeatedly in Equation (14), with initial value $CT^1(s_l) = PT^1(s_l)$. α is a smoothing factor such that $0 < \alpha < 1$.

In consequent expansion of Equation (14), weights assigned to PT values, decrease exponentially from the most recent PT value to the PT values at earlier time instances. At the same time, it achieves uniform effect in the prediction of cumulative trust, by tuning of smoothing factor α. We recommend that the smoothing factor α should be tuned to a value in the range from 0.1 to 0.4. This enables predicted cumulative trust to match closely with the actual trust value.

5 Algorithm for Trust Estimation

Algorithm 1 shows the steps for adaptive and dynamic trust estimation of a cloud service. The trust assessments are performed during the interaction between a user and the service, based on service evidence factors. The algorithm takes a set of cloud service attributes and a number of time instances as input for trust computation of a service. Threshold trust (TH) taken as another input, is the minimum expected trust by the user. The algorithm takes historical trust value and user ratings as input for trust initialization of a cloud service. The algorithm gives the output as sets of present trust and cumulative trust values for service s_l over the period of interaction. Historical trust value (HS) indicates the cumulative trust value of the cloud service saved as a result of the last interaction of the user with the service. The steps of Algorithm 1 are explained as below.

Step 1. (Line 4) At the start of the user interaction with the service, trust is initialized as follows:

i) If the user has interacted with the service before, the trust is initialized to a value as indicated by *HS*.

ii) Otherwise initial trust is set based on the interactions of other users with the service. Here, the ratings for the service, given by other users are used for trust initialization. Computation of reputation score using beta probability density function is described by Josang *et al.* [12]. Here, in Algorithm 1, the same notion of beta

Algorithm 1 Trust Estimation for cloud service s_l

```
 1: Input:
      a. Set of $m$ cloud service attributes,
         $(AC) = \{R_1, R_2, ..., R_m\}$
      b. Number of time instances $(n)$
      c. Threshold trust $(TH)$
         // minimum expected trust value
      d. Historical trust value $(HS)$
      e. User ratings $(ER_l) = \{p_l, n_l\}$
         // number of positive and negative ratings
         // for service $s_l$
 2: Output:
      a. Set of Present Trust values for service $s_l$,
         $LP = \{PT[1], PT[2], ..., PT[n]\}$
      b. Set of Cumulative Trust values for service $s_l$,
         $LC = \{CT[1], CT[2], ..., CT[n]\}$
 3: Begin
 4: $IT = Trust\_initiate(s_l, HS, ER_l)$;
         // Trust initialization for service $s_l$
 5: if $IT \geq TH$ then
 6:    $Matrix\ C = Get\_evidences(s_l, AC, n)$;
 7:    $Matrix\ M = Normalize\_evidences(C, AC)$;
         // Function $f_{NE}$ in Definition 1
 8:    $Set\ W = Compute\_weights(M, AC, n)$;
         // $W$ is a set of weights of $m$ attributes,
         // computed by Algorithm 2,
         // $W = \{w_1, w_2, ..., w_m\}$
 9:    $i = 1$;
10:    while $i \neq n$ do
11:       Compute Present Trust of service $s_l$ at
          time instant $i$ as: $PT[i] = \sum_{j=1}^{m} w_j h_{ij}$;
            // Function $f_{PT}$ in Definition 1
            // From Algorithm 2, $w_j$ is a weight
            // and $h_{ij}$ is an element of matrix $M$
12:       Add $PT[i]$ in set $LP$;
13:       if $i = 1$ then
14:          $CT[i] = PT[i]$;
15:       else
16:          Compute Cumulative Trust of service $s_l$
             as: $CT[i] = \alpha PT[i] + (1 - \alpha)CT[i - 1]$;
             // Function $f_{CT}$ in Definition 1
             // $\alpha$ is a smoothing factor, $0 < \alpha < 1$
17:       end if
18:       Add $CT[i]$ in set $LC$;
19:       if $CT[i] < TH$ then
20:          Notify incident;
21:       end if
22:       $Update\_trust(s_l, CT[i])$;
23:       $i = i + 1$;
24:    end while
25: end if
26: End
```

probability density function is used to compute the Initial Trust (IT) and is given by:

$$IT = \frac{p_l + 1}{p_l + n_l + 2} \quad (15)$$

where p_l and n_l are number of positive and negative ratings about the service s_l as shown by the set ER_l. The individual rating submitted by the user is in the range $[0, 1]$. A rating greater than or equal to 0.5 is considered as positive rating and a rating less than 0.5 is treated as a negative rating.

iii) When ratings about the service are not available, then IT becomes equal to 0.5 as deduced by Equation (15).

If the initial trust satisfies the threshold trust requirement, then the algorithm proceeds with the next steps.

Step 2. (Line 6) Trust estimator gets the evidence factors of the cloud service and gives the resultant evidence matrix C as shown in Equation (1).

Step 3. (Line 7) Normalization function takes the evidence matrix as input and transforms values of all the attributes in the matrix to uniform range as specified by Equations (2) and (3). It results into the normalized evidence matrix M as depicted by Equation (4).

Step 4. (Line 8) Here, the algorithm invokes the function to perform adaptive computation of weights for attributes of the cloud service. The details of the function to compute weights are given by Algorithm 2 in Section 5.1.

Step 5. (Lines 9 - 24) At each instant of time, adaptive computation of present trust is performed using the normalized evidence factors and the weights of service attributes. Subsequently, cumulative trust, representing the dynamic trust prediction of a service, is computed. The output sets of present trust and cumulative trust values are populated with the corresponding computed trust values. At any time, if cumulative trust falls below TH, then the user is notified about the incident. At each time instant, the HS of the service is revised by the latest computed cumulative trust value. Computations of present trust and cumulative trust values are described in Section 4 with elaboration.

5.1 Algorithm for Computation of Weights

Algorithm 2 describes the steps for adaptive computation of weights for attributes of the cloud service. The algorithm takes a normalized evidence matrix, a set of cloud service attributes and a number of time instances as input. The algorithm in turn, gives the set of weights for the attributes of a cloud service as the output. As shown in the algorithm, average of the evidence factors, variation factor and the impact of variation factor are computed for each attribute. From the values of impact of variation factor, weight of each attribute is computed. The computed weight of each attribute is added to the output set of weights. The details of computation of adaptive weights are presented in Section 4.2.

Algorithm 2 Computation of weights for the attributes of a cloud service: *Compute_weights(M,AC,n)*

```
Input:
    a. Matrix $M = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & h_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \dots & h_{nm} \end{bmatrix}$
       // Normalized evidence matrix
    b. Set of $m$ cloud service attributes,
       $(AC) = \{R_1, R_2, ..., R_m\}$
    c. Number of time instances $(n)$
Output: Set of weights of m attributes,
       $W = \{w_1, w_2, ..., w_m\}$
Begin
$sum = 0$;
foreach $R_j \in AC$ do
    Compute average of evidence factors as:
        $A(R_j) = (\sum_{i=1}^{n} h_{ij})/n$;
    Calculate variation factor as:
        $V(R_j) = \sum_{i=1}^{n} (h_{ij} - A(R_j))^2$;
    Calculate impact of variation factor as:
        $F[j] = 1/(V(R_j) + (1/n))$;
    $sum = sum + F[j]$;
end
$j = 1$;
while $j \neq m$ do
    Compute weight of an attribute $R_j$ as:
        $w_j = F[j]/sum$;
        // Function $f_{CW}$ in Definition 1
    Add $w_j$ in set $W$;
    $j = j + 1$;
end while
End
```

5.2 Computational Complexity of Trust Estimation

The details of computational complexity of Algorithm 1 are described as follows. The computational complexity of a function to get the evidence factors is $O(mn)$. The computational complexity of normalization operation is $O(mn)$. Computation of present trust has the complexity of $O(nm)$. The computational complexity of cumulative trust prediction is $O(n)$. Computational complexity of a function to update trust is $O(n)$. As shown by the steps in Algorithm 2, computation of weights has the complexity of $O(mn^2)$. Therefore, the overall computational complexity (CC) of trust estimation (including Algorithm 1 and Algorithm 2) is given as below:

$$CC = O(mn) + O(n) + O(mn^2) = O(mn^2) \qquad (16)$$

Thus, the overall computational complexity of trust estimation depends on the number of cloud service attributes (m) and the number of time instances (n).

6 Performance Evaluation

For the evaluation of trust estimation model EBTEM described in Section 4, a prototype is developed in Java. The general and the most relevant attributes of a cloud service, as discussed in Section 2, which include availability, throughput, response time and security, are used during the experimentation. These attributes are important as they reflect the ability of a cloud service to perform the various operations effectively. For the values of throughput (kbps) and response time (seconds), real world QoS data set [30] is referred. The availability implies the percentage of time the cloud service is accessible. Security attribute is considered as the percentage of the number of violation incidents related to authentication or authorization. Weibull distribution is the suitable theoretical distribution for modeling failure time and can also be employed for modeling inputs in the absence of real data [14]. Hence, values of availability (%) and security violation incidents (%) are generated using the Weibull distribution. Various values of the attributes are normalized in the range [0.01, 0.99]. For a cloud service, higher values of availability and throughput are desired. Hence, values of these attributes are normalized using Equation (2). Whereas, lower values of response time and security violation incidents are expected. Hence, values of these attributes are normalized using Equation (3).

6.1 Trust Models for Comparison

In addition to our trust model EBTEM, two other trust models have also been implemented for comparative assessment. They are, averaging based simple trust model (ASTM) and weighted summation based trust model (WSTM). Both, ASTM and WSTM make use of multiple cloud QoS attributes for trust evaluation. These trust models are selected to compare the performance of EBTEM from two perspectives: i) To compare with the model where weights assigned to the various factors for trust assessment are static and subjective in nature. Thus, ASTM represents this trust model where equal weights are assigned to all the factors. ASTM is analogous to the model proposed by Manuel [17]. ii) To compare with the other model where dynamic weights are assigned to all the factors for trust assessment. WSTM represents this trust model and it corresponds to the model proposed by Li *et al.* [15]. The trust models [15, 17] are discussed in Section 2.

In ASTM, present trust (PT) of a cloud service (s_l) is computed as an average of all m evidence factors at a time instant i. It is given by:

$$AT^i(s_l) = (\sum_{j=1}^{m} h_{ij})/m \tag{17}$$

where h_{ij} is a normalized evidence factor of attribute R_j at time instant i. Cumulative Trust (CT), at time instant n, is calculated as an average of PT values, given as:

$$T^n(s_l) = (\sum_{i=1}^{n} AT^i(s_l))/n \tag{18}$$

where $AT^i(s_l)$ is a PT at time instant i.

In WSTM, PT of a cloud service (s_l) is computed as a weighted summation of all m evidence factors at time instant i. It is given by:

$$DT^i(s_l) = \sum_{j=1}^{m} w_j^{'} h_{ij} \tag{19}$$

where h_{ij} is a normalized evidence factor of attribute R_j at time instant i and $w_j^{'}$ is a weight assigned to it such that $0 < w_j^{'} < 1$ and $\sum_{j=1}^{m} w_j^{'} = 1$. Here, weights are computed using an approach of information entropy based weights, proposed by Li *et al.* [15]. CT at time instant n, is given by:

$$IT^n(s_l) = \sum_{i=1}^{n} w_i^{''} DT^i(s_l) \tag{20}$$

where $DT^i(s_l)$ is a PT of a cloud service (s_l) at time instant i and $w_i^{''}$ is a weight assigned to it such that $0 < w_i^{''} < 1$ and $\sum_{i=1}^{n} w_i^{''} = 1$. Here, weights are computed using the approach proposed by Li *et al.* [15] for assigning weight to each Real-Time Trust Degree (RTD). Here decreasing weights are assigned from latest RTD to RTDs at previous time instances.

6.2 Evaluation Metrics

The effectiveness of trust assessment method depends on the accuracy of trust estimation. Mean Absolute Error (MAE) [3] is a metric to assess an error in the prediction process. Here it is devised to compute an error in the prediction of cumulative trust and thus, to analyze the accuracy of trust estimation. Consequently, MAE is formulated as below.

$$MAE = \frac{1}{n}(\sum_{i=1}^{n} |P^{i+1}(s_l) - C^i(s_l)|) \tag{21}$$

where $P^{i+1}(s_l)$ is present trust of a cloud service (s_l) at time instant $(i+1)$, $C^i(s_l)$ is a predicted cumulative trust of a cloud service (s_l) at time instant i and n is the total number of time instances for assessment of MAE. Smaller value of MAE indicates higher accuracy of trust estimation and hence better performance of the trust model.

MAE as an absolute error based measure is complemented by a measure based on relative percentage error to analyze accuracy of trust estimation. Symmetric Mean Absolute Percentage Error (SMAPE) [13] is such a measure of error. SMAPE is formulated as below.

$$SMAPE = \frac{(\sum_{i=1}^{n} |C^i(s_l) - P^{i+1}(s_l)|) \times 100}{\sum_{i=1}^{n}(P^{i+1}(s_l) + C^i(s_l))} \tag{22}$$

where $P^{i+1}(s_l)$ is present trust of a cloud service (s_l) at time instant $(i+1)$, $C^i(s_l)$ is a predicted cumulative trust of a cloud service (s_l) at time instant i and n is the total number of time instances for assessment of SMAPE. From Equation (22), it enables to evaluate unbiased assessment of error in the prediction process. Similar to MAE, a smaller value of SMAPE signifies that predicted trust values closely match with actual trust values, leading to better accuracy of trust estimation.

Along with the accuracy in the calculation, trust model should be able to accomplish the task of rapid trust assessment. This is essential to service the upcoming requests for trust assessment efficiently. Hence, Mean Execution Time (MET) is used as a metric to evaluate the computational efficiency of trust assessment.

6.3 Results and Analysis

The performance of our trust model EBTEM is evaluated in terms of the accuracy and computational efficiency of trust estimation.

6.3.1 Assessment of Accuracy

The comparative evaluation of accuracy of trust estimation is performed during three sets of experiments, by observing MAE and SMAPE values of three trust models. The number of evidence samples is varied from 50 to 500, in all the sets of experiments. The comparisons of error and accuracy are made in all the sets of experiments, at a mid-point of sample counts, which is 250. The results show less error and better accuracy for EBTEM as against the ASTM and WSTM. The details are explained below. For other sample counts, the results may vary within marginal limits. However, the consistency of better accuracy of EBTEM remains more or less the same, in comparison with ASTM and WSTM, for any number of attributes of a cloud service.

In the first set of experiments, evidence samples for the two attributes which include throughput and response time, are considered. The results in Figure 3 illustrate that, the MAE of EBTEM is 0.035 whereas for ASTM it is 0.144 and for WSTM, it is 0.166. Thus, MAE of ASTM is 4.11 times the one in EBTEM and that of WSTM is 4.74 times the one in EBTEM. This implies that the accuracy of EBTEM is higher than ASTM by 11.3% and is higher than WSTM by 13.6%. As seen in Figure 4, the SMAPE of EBTEM is 1.85 whereas for ASTM it is 9.06 and for WSTM it is 10.85. Thus, SMAPE of ASTM is 4.9 times the one in EBTEM and that of WSTM is 5.86

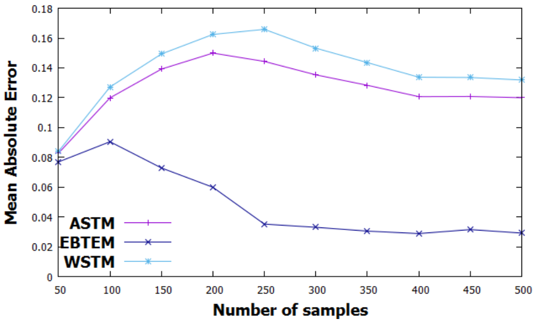


Figure 3: MAE with two cloud service attributes

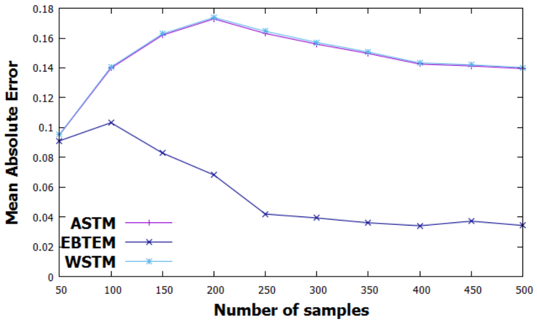


Figure 5: MAE with three cloud service attributes

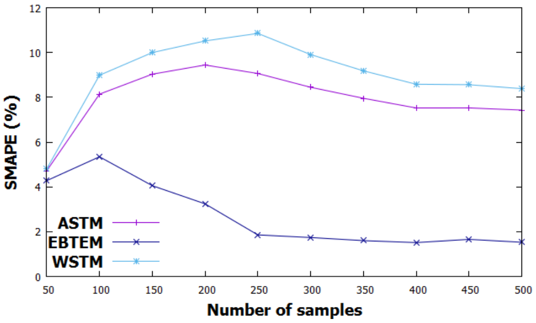


Figure 4: SMAPE with two cloud service attributes

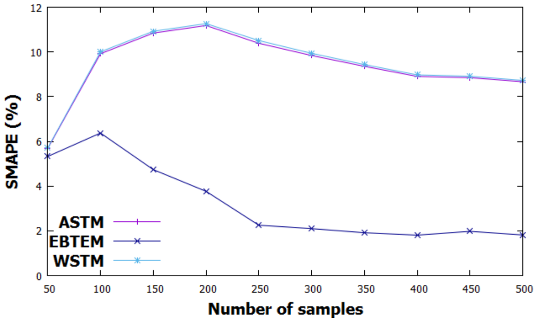


Figure 6: SMAPE with three cloud service attributes

times the one in EBTEM. This signifies that the accuracy of EBTEM is higher than ASTM by 7.35% and is higher than WSTM by 9.17%. Thus, the results in Figures 3 and 4 signify that, the performance of EBTEM is better than that of the other models for various number of samples of two attributes.

For the second set of experiments, evidence samples for the three attributes which cover availability, throughput and response time, are taken. The results in Figure 5 show that, the MAE of EBTEM is 0.041 whereas for ASTM it is 0.163 and for WSTM, it is 0.165. Thus, MAE of ASTM is 3.98 times the one in EBTEM and that of WSTM is 4.02 times the one in EBTEM. This implies that accuracy of EBTEM is higher than ASTM by 12.72% and is higher than WSTM by 12.93%. From Figure 6, the SMAPE of EBTEM is 2.24 whereas for ASTM it is 10.39 and for WSTM it is 10.51. Thus, SMAPE of ASTM is 4.64 times the one in EBTEM and that of WSTM is 4.69 times the one in EBTEM. This signifies that accuracy of EBTEM is higher than ASTM by 8.34% and is higher than WSTM by 8.46%. Thus, the results in Figures 5 and 6 depict that, with the evidences of three cloud service attributes as well, the performance of EBTEM is better than that of the other models for various number of samples.

The third set of experiments makes use of evidence samples for the four attributes which incorporate availability, throughput, response time and security violation incidents. The results in Figure 7 show that, the MAE of EBTEM is 0.045 whereas for ASTM it is 0.126 and for WSTM, it is 0.127. Thus, MAE of ASTM is 2.8 times the one in EBTEM and that of WSTM is 2.82 times the one in EBTEM. This implies that the accuracy of EBTEM is higher than ASTM by 8.48% and is higher than WSTM by 8.59%. From Figure 8, the SMAPE of EBTEM is 2.53 whereas for ASTM it is 8.43 and for WSTM it is 8.46. Thus, SMAPE of ASTM is 3.33 times the one in EBTEM and that of WSTM is 3.34 times the one in EBTEM. This signifies that accuracy of EBTEM is higher than ASTM by 6.05% and is higher than WSTM by 6.08%. Thus, the results in Figures 7 and 8 demonstrate that, with the evidences of four cloud service attributes as well, the performance of EBTEM is better than that of the other models for various number of samples.

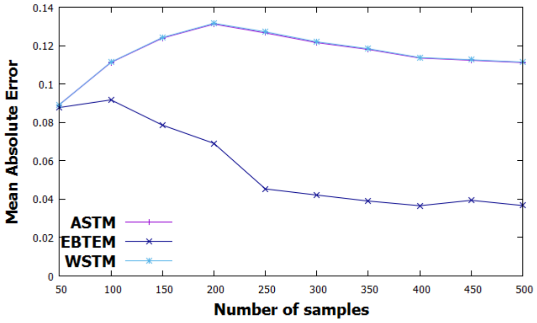


Figure 7: MAE with four cloud service attributes

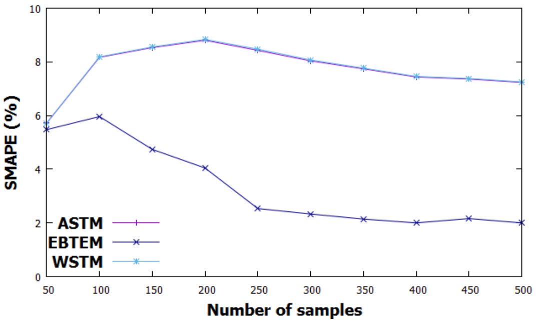


Figure 8: SMAPE with four cloud service attributes

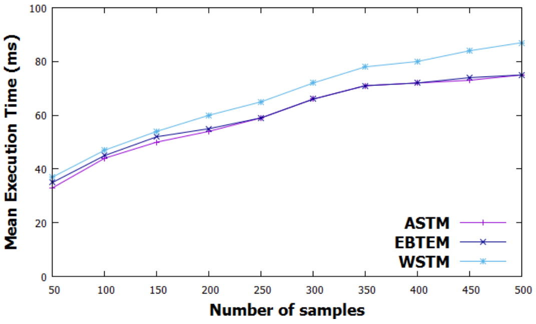


Figure 9: MET with two cloud service attributes

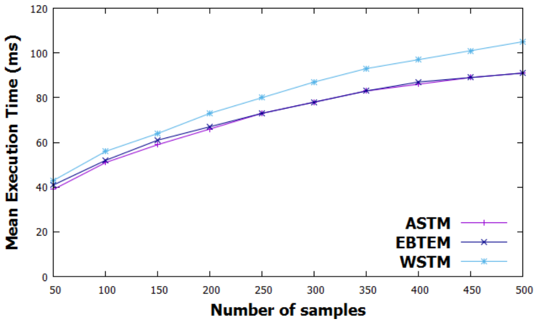


Figure 10: MET with three cloud service attributes

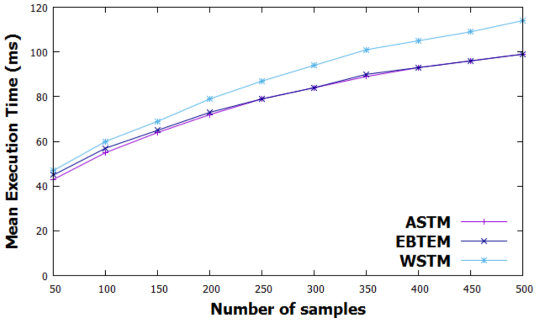


Figure 11: MET with four cloud service attributes

The results in Figure 3 to Figure 8 show that, with the increase in the number of samples, MAE and SMAPE values of EBTEM are decreased. Moreover, even with a smaller number of samples, EBTEM depicts higher accuracy and hence signifies the better performance. This makes EBTEM more suitable for cloud-based deployments in practice where, with limited number of samples, trust estimation can be performed efficiently.

6.3.2 Assessment of Computational Efficiency

The comparative assessment of efficiency of trust computation is performed with a set of three experiments, by observing MET values of three trust models. Experimentation is carried out using a computer with Intel i7-3537, 2.00 GHz CPU and 8 GB RAM. The number of evidence samples is varied from 50 to 500, in all the experiments. Figure 9 shows the results of the first experiment, where evidence samples for the two attributes which include throughput and response time, are considered. Figure 10 depicts the MET values of the second experiment, where evidence samples for the three attributes which cover availability, throughput and response time, are taken. Figure 11 demonstrates the results of the third experiment, where evidence samples for the four attributes which incorporate availability, throughput, response time and security violation incidents, are used.

Results in Figures 9, 10 and 11 show that, initially when the number of samples is reasonably small, observed MET values of the three models are nearby to each other. For instance, from Figure 9, at a sample count of 150, MET of EBTEM is 52, for ASTM it is 50 and for WSTM it is 54. When the number of samples becomes large, MET values of EBTEM and that of ASTM increase along smooth curves and follow the same trend in which the values of MET closely match with each other. On the other hand, MET values for WSTM increase and deviate as compared to the other two models. For example, from Figure 10, at a sample count of 450, MET value reaches to 89 for both, EBTEM as well as ASTM and for WSTM it is 101. Similarly, for instance, from Figure 11, at a sample count of 500, MET value is 99, for both, EBTEM as well as ASTM and for WSTM it is 114. Thus, it can be noted from Figures 9, 10 and 11, that ASTM has the lowest MET values and MET values for EBTEM almost match with those of ASTM. WSTM has comparatively greater MET values. In a nutshell, EBTEM and ASTM demonstrate better computational efficiency.

Although, ASTM coincides with EBTEM along the dimension of efficiency, accuracy of EBTEM is the highest and that of ASTM is much low. This is indicated by the results in Figure 3 to Figure 8. Hence, our trust model EBTEM exhibits much better performance. For the purpose of experimentation and performance analysis, we used the sets of two, three and four cloud service attributes in trust estimation. However, as elaborated in Section 4, the methodology of trust model enables to take into account multiple cloud service attributes for trust assessment.

7 Conclusions

In this paper, we presented the evidence based trust estimation model (EBTEM) and the algorithm for trust estimation of a cloud service. Adaptive and dynamic trust assessments are the main facets offered by the model. The evidence factors of a cloud service are aggregated to compute its present trust by assigning adaptive weights to the attributes of the service. The model enables continuous trust assessment of a cloud service according to the real-time changing values of the service attributes. Thus, the cloud user can decide whether to continue using the cloud service based on cumulative trust indicated by the model.

Experimental results have shown that average Mean Absolute Error (MAE) of averaging based simple trust model (ASTM) and weighted summation based trust model (WSTM) are respectively 3.63 and 3.86 times that of EBTEM. Hence, based on MAE, the average accuracy of EBTEM is higher than ASTM by 10.83% and by 11.71% than WSTM. Also, average Symmetric Mean Absolute Percentage Error (SMAPE) of ASTM and WSTM are respectively 4.29 and 4.63 times that of EBTEM. Hence, with regard to SMAPE, the average accuracy of EBTEM is higher than ASTM by 7.25% and by 7.9% than WSTM. Results have also demonstrated that, even with the limited number of samples, EBTEM achieves higher accuracy. Thus, results have shown that performance of EBTEM is much better than that of other models for the dimensions of accuracy and computational efficiency of trust assessment. In conclusion, our trust model EBTEM depicts effective trust estimation of a cloud service. As future work, we aim to extend our trust model by taking into account Quality of Service (QoS) related requirements of the user for assessment of trust.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments. Sincere thanks are also due to College of Engineering Pune and Cummins College of Engineering for Women, Pune, India.

References

[1] J. Abawajy, "Establishing trust in hybrid cloud computing environments," in *10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'11)*, pp. 118–125, 2011.

[2] I. M. Abbadi and A. Martin, "Trust in the cloud," *Information Security Technical Report*, vol. 16, no. 3, pp. 108–114, 2011.

[3] C. Chatfield, *Time-series Forecasting*, CRC Press, 2000.

[4] W. J. Fan, S. L. Yang, H. Perros, and J. Pei, "A multi-dimensional trust-aware cloud service selection mechanism based on evidential reasoning approach," *International Journal of Automation and Computing*, vol. 12, no. 2, pp. 208–219, 2015.

[5] N. Ghosh, S. K. Ghosh, and S. K. Das, "Selcsp: A framework to facilitate selection of cloud service providers," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 66–79, 2015.

[6] S. M. Habib, S. Hauke, S. Ries, and M. Mühlhäuser, "Trust as a facilitator in cloud computing: A survey," *Journal of Cloud Computing*, vol. 1, no. 1, pp. 1–18, 2012.

[7] S. M. Habib, S. Ries, and M. Mühlhäuser, "Cloud computing landscape and research challenges regarding trust and reputation," in *Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC'10)*, pp. 410–415, 2010.

[8] S. M. Habib, S. Ries, and M. Mühlhäuser, "Towards a trust management system for cloud computing," in *10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'11)*, pp. 933–939, 2011.

[9] W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing.," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[10] J. Huang and D. M. Nicol, "Trust mechanisms for cloud computing," *Journal of Cloud Computing*, vol. 2, no. 1, pp. 1–14, 2013.

[11] Y. Huo, Y. Zhuang, and S. Ni, "Fuzzy trust evaluation based on consistency intensity for cloud services," *Kybernetes*, vol. 44, no. 1, pp. 7–24, 2015.

[12] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[13] V. Kreinovich, H. T. Nguyen, and R. Ouncharoen, "How to estimate forecasting quality: A system-motivated derivation of symmetric mean absolute percentage error (smape) and other similar characteristics," 2014.

[14] A. M. Law, *Simulation modeling and analysis*, McGraw-Hill, 2007.

[15] X. Li, H. Ma, F. Zhou, and X. Gui, "Service operator-aware trust scheme for resource matchmaking across multiple clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1419–1429, 2015.

[16] Z. Li, L. Liao, H. Leung, B. Li, and C. Li, "Evaluating the credibility of cloud services," *Computers & Electrical Engineering*, 2016.

[17] P. Manuel, "A trust model of cloud computing based on quality of service," *Annals of Operations Research*, pp. 1–12, 2013.

[18] P. D. Manuel, M. I. Abd-El Barr, and S. T. Selvi, "A novel trust management system for cloud computing iaas providers," *Journal of Combinatorial Mathematicsand Combinatorial Computing*, vol. 79, p. 3, 2011.

[19] P. D. Manuel, S. T. Selvi, and M. I. Abd-El Barr, "Trust management system for grid and cloud resources," in *First International Conference on Advanced Computing (ICAC'09)*, pp. 176–181, 2009.

[20] A. Mosa, H. M. El-Bakry, S. M. Abd El-Razek, S. Q. Hasan, "A proposed E-government framework based on cloud service architecture," *International Journal of Electronics and Information Engineering*, vol. 5, no. 2, pp. 93–104, 2016.

[21] F. Moyano, K. Beckers, and C. Fernandez-Gago, "Trust-aware decision-making methodology for cloud sourcing," in *26th International Conference on Advanced Information Systems Engineering (CAiSE'14)*, pp. 136–149, 2014.

[22] T. H. Noor and Q. Z. Sheng, "Credibility-based trust management for services in cloud environments," in *9th International Conference on Service-Oriented Computing (ICSOC'11)*, pp. 328–343, 2011.

[23] T. H. Noor and Q. Z. Sheng, "Trust as a service: a framework for trust management in cloud environments," in *12th International Conference on Web Information System Engineering (WISE'11)*, pp. 314–321, 2011.

[24] T. H. Noor, Q. Z. Sheng, S. Zeadally, and J. Yu, "Trust management of services in cloud environments: Obstacles and solutions," *ACM Computing Surveys*, vol. 46, no. 1, pp. 12, 2013.

[25] D. Patel, "Accountability in cloud computing by means of chain of trust dipen contractor," *International Journal of Network Security*, vol. 19, no. 2, pp. 251-259, 2017.

[26] P. S. Pawar, M. Rajarajan, S. K. Nair, and A. Zisman, "Trust model for optimized cloud services," in *Trust Management VI*, pp. 97–112, 2012.

[27] C. Qu and R. Buyya, "A cloud trust evaluation system using hierarchical fuzzy inference system for service selection," in *28th International Conference on Advanced Information Networking and Applications (AINA'14)*, pp. 850–857, 2014.

[28] J. Sidhu and S. Singh, "Improved topsis method based trust evaluation framework for determining trustworthiness of cloud service providers," *Journal of Grid Computing*, pp. 1–25, 2016.

[29] W. L. Tai, Y. F. Chang, "Comments on a secure authentication scheme for IoT and cloud servers," *International Journal of Network Security*, vol. 19, no. 4, pp. 648-651, 2017.

[30] Y. Zhang, Z. Zheng, and M. R. Lyu, "Wspred: A time-aware personalized qos prediction framework for web services," in *22nd International Symposium on Software Reliability Engineering (ISSRE'11)*, pp. 210–219, 2011.

Biography

Shilpa Deshpande is a research scholar at College of Engineering Pune, Savitribai Phule Pune University, India. Her research interests are in the area of cloud computing and distributed systems. She is currently an Assistant Professor with the Computer Engineering Department, Cummins College of Engineering for Women, Pune. She has received the Bachelor's degree and the Master's degree in Computer Engineering from Savitribai Phule Pune University, in 1996 and in 2002 respectively.

Rajesh Ingle is an Adjunct Professor at Department of Computer Engineering, College of Engineering Pune, India. He is a Professor at Department of Computer Engineering, Pune Institute of Computer Technology, Pune. He has received Ph.D. Computer Science and Engineering from Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Powai, Mumbai. His research interests include Distributed system security and Cloud security. He has received the B.E. Computer Engineering from Pune Institute of Computer Technology and M.E. Computer Engineering from Government College of Engineering, Savitribai Phule Pune University. He has also received M.S. Software Systems from BITS, Pilani, India. He is a senior member of the IEEE, IEEE Communications Society and IEEE Computer Society.

Power Efficient Secure Web Servers

Sujatha Sivabalan, P. J. Radcliffe
(Corresponding author: Sujatha Sivabalan)

The Department of Electrical and Computer Engineering& RMIT University
Swanston Street, Melbourne, Australia
(Email: s3365148@student.rmit.edu.au)

(Received Feb. 14, 2017; revised June 5, 2017)

Abstract

The power consumption of web servers and associated security devices is becoming an increasing issue both from an economic and environmental perspective. This paper analyses the power consumption of both security software and web server software and concludes that traditional architectures waste energy with repeated transitions up and down the TCP/IP stack. This contention is proved by comparing the energy usage of a traditional architecture and a new architecture whereby IDS functionality is moved into the web server and all operations share HTTP packets. Based on these findings we propose a novel alternative power efficient architecture for web servers that may also be usable in other network systems.

Keywords: Central Processing Unit CPU; Intrusion Detection System IDS; Intrusion Prevention System IPS; Web Server

1 Introduction

our world is steadily accelerating towards an Internet based economy where web servers are most significant. In an enterprise or commercial data centre, the power usage of web servers [13, 26, 29] is becoming a major concern, particularly where the web servers need to handle a heavy traffic load or may be subject to Denial of Service (DoS) attack [38]. It is desirable to reduce the power consumption in such systems both from an environmental and economic point of view. Furthermore, devices that can cope with high traffic loads are very expensive as well as consuming considerable power [6, 17, 18, 20, 22, 35, 37]. The web server system in a web server, or web server farm, consumes power for different reasons. Such systems use power to provide web services for the ingress traffic and for a number security functions that may be separate boxes or virtual machines.

The goal of this paper is to analyse the power usage of the security software or devices within the system. Such a comparison is traditional impossible as the software resides on a variety of boxes which each box having different internal hardware and CPU types. Even in a cloud environment, a software application such as an IDS may be run on different types of CPUs and this will confuse any attempt to measure power consumption. The exception here is Software Defined Networks (SDN) where security applications can all be run on the one hardware platform [40]. This approach is used to achieve the goals of this paper, all programs can be run on the same hardware and their power consumption compared by measuring CPU utilization.

This paper focuses on IDS/IPS systems as they have the most scope to be absorbed into other devices. Two well-known and widely used IDS/IPS programs were selected for testing, Bro [8] and Snort [31]. Bro [21] is capable of sophisticated packet analysis and a full IDS function. Snort [1, 3, 9, 15, 21] has some IPS capability but it is essentially an IDS where detection based on packet signature matching.

The key research outcomes described in this paper include:

- Power consumption analysis of two traditional security applications Bro and Snort.
- Power consumption analysis of a new daemon developed [27, 28] by us that integrates into Apache.
- Based on the findings, a novel alternative architecture for webservers is proposed that can reduce total power usage.

This paper organized as follows: Section 2 reviews web server power consumption and discusses existing work related to reducing IDS and security device power consumption. Section 3 measures and analyses the power consumption of BRO and Snort. Section 4 extends the experiments by replacing Bro and Snort with an IDS daemon that works with Apache. Section 5 uses the findings from the experiments to propose a novel architecture that reduces CPU load and hence power consumption. Section 6 provides conclusions to the work and offers some further research directions.

2 Related Work

2.1 CPU Utilization and Power Consumption

Power management of web servers systems is an area of ongoing research. Sharma *et al.* [26] states that power consumption on a web server is economically and ecologically significant. In 2002 Bohrer *et al.* [7] started focussing on managing power consumption in web servers and their experimental results shows that the CPU consumes the largest fraction of the system Power. Mukherjee *et al.* [24] experimentally observed that Disk and Memory I/O usage had a negligible effect on server power consumption, whereas the CPU utilization is linearly related to power consumption. Given these observations, it is important to minimize CPU utilization in order to reduce power consumption. Minas *et al.* [23] mentioned that CPU usage in a web server is the most significant factor in power consumption and his results on a quad core Intel processor shows that for a given CPU, the power consumption is linearly related to the CPU utilization. Other processors have a similar relationship. From these observations, we can conclude that it is appropriate to use CPU utilization as a proxy for power consumption of software providing that the CPU type and hardware is the same. A weakness for evaluating the power usage of several devices is that the CPU can vary between physical devices or even virtual machines in the cloud and so cumulative CPU utilization is not a good measure of power usage.

2.2 IDS Power Consumption

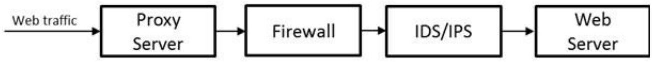


Figure 1: Web server in a Host-based system.

In a host-based system as shown in Figure 1, devices such as proxy servers, firewalls, and Intrusion Detection/Prevention systems (IDS/IPS) secure web servers. These services are crucial as they can eliminate or reduce attack traffic. It is notable that in Figure 1 the traffic is received, analysed, and transmitted three times before getting to the web server. Only the analysis activity has value with the repeated reception and transmission being an overhead that consumes power. The IDS function is of particular interest as if it is found to be inefficient in terms of power use then its functionality might be able to be absorbed into other devices. There is significant research on ways to improve the performance of IDS/IPS that reduces CPU load and hence power consumption. Zaman *et al.* [39] implemented a light weight IDS to overcome resource consumption but this achieved poor detection rate based on two different approaches. Wheeler *et al.* [37] introduced three levels of parallelism using node, component and sub component level and stateless analysis but this achieved little in the way of power savings.

Vasiliadis *et al.* [35] implemented a multi-parallel IDS architecture (MIDEA) for high-performance processing and stateful analysis of network traffic and while this worked well it is expensive and complex to implement [9]. The major flaw of these [35, 37, 39] methods is repetition [9] where one packet undergoes the reception, inspection, and transmission several times hence there is waste of CPU computing time and electrical power. Waleed *et al.* [9] suggested parallel NIDS to reduce packet drop rate and process more packets in less time during heavy traffic, however sudden increases in traffic will result in packet loss. This approach requires many IDS in parallel that in turn increases the CPU load and leads to increased power consumption and cost. Several researchers [16,26-30] have used various methods to implement a lightweight IDS approach but the overall result has been to increase the workload on the web server. The methods reviewed all have at least one of two flaws that contribute to power consumption:

- There is multiple reception, analysis, and retransmission of HTTP packets which costs computing time and electrical power.

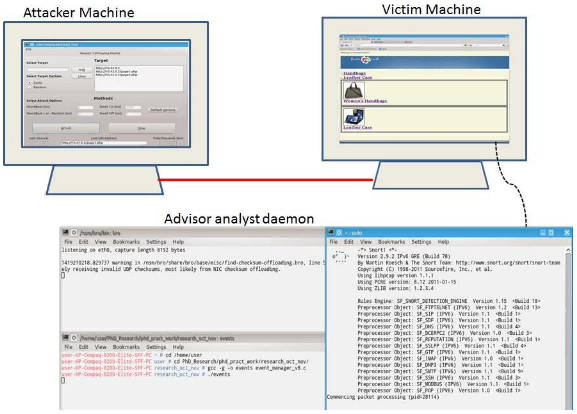
- All the security devices placed at the network ingress point must handle the full load of normal and DDoS traffic thus requiring powerful, power hungry and expensive devices.

3 Power Analysis of Traditional Approaches: BRO and SNORT

3.1 Experimental Setup

This section describes the experimental set up used by the authors to measure the CPU load used by Bro and Snort and then discusses the result.

The two experimental setups depicted in Figure 2 & Figure 3, both used i7 desktop computers running Linux Mint 13. Figure 2 illustrates the low traffic test bed that used only two systems; one serves as the victim machine with the detection functionality and other serves as the attacker machine. Figure 3 illustrates the high traffic test bed that used twenty computers, one was the victim web server plus the IDS functionality and the others servers acted as the zombie machines that attacked the victim. The CPU load measurement is the load on one CPU of a multi-core processor and it is independently measured for both the web server and IDS function. The server on the victim machine was an Apache 2 web server with three web pages including images. The attacker machines generate random DDoS traffic aimed at these web pages. Each IDS only sees the incoming web page requests as the outgoing web pages come from the system's server and are assumed safe.

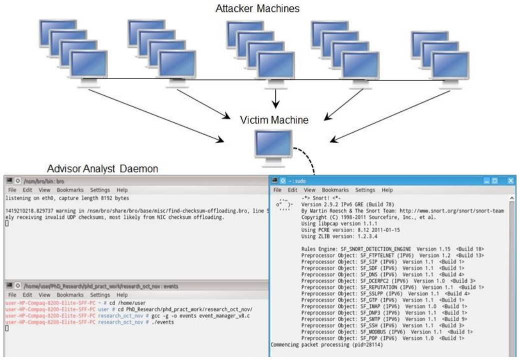


```
listening on eth0, capture length 8192 bytes

1419210218.829737 warning in /nsm/bro/share/bro/base/misc/find-checksum-offloading.bro, line 5
ely receiving invalid UDP checksums, most likely from NIC checksum offloading.
```

```
user-HP-Compaq-8200-Elite-SFF-PC ~ # cd /home/user
user-HP-Compaq-8200-Elite-SFF-PC user # cd PhD_Research/phd_pract_work/research_oct_nov/
user-HP-Compaq-8200-Elite-SFF-PC research_oct_nov # gcc -g -o events event_manager_v8.c
user-HP-Compaq-8200-Elite-SFF-PC research_oct_nov # ./events
```

```
   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.2 IPv6 GRE (Build 78)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 1998-2011 Sourcefire, Inc., et al.
           Using libpcap version 1.1.1
           Using PCRE version: 8.12 2011-01-15
           Using ZLIB version: 1.2.3.4

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 1.15  <Build 18>
           Preprocessor Object: SF_FTPTELNET (IPV6)  Version 1.2  <Build 13>
           Preprocessor Object: SF_SIP (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS (IPV6)  Version 1.1  <Build 4>
           Preprocessor Object: SF_DCERPC2 (IPV6)  Version 1.0  <Build 3>
           Preprocessor Object: SF_REPUTATION (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP (IPV6)  Version 1.1  <Build 4>
           Preprocessor Object: SF_GTP (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP (IPV6)  Version 1.0  <Build 1>
           Preprocessor Object: SF_DNP3 (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SMTP (IPV6)  Version 1.1  <Build 9>
           Preprocessor Object: SF_SSH (IPV6)  Version 1.1  <Build 3>
           Preprocessor Object: SF_MODBUS (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP (IPV6)  Version 1.0  <Build 1>
Commencing packet processing (pid=28114)
```

Figure 2: Experimental set up low attack traffic


Attacker Machines
Victim Machine
Advisor Analyst Daemon
/nsm/bro/bin: bro
/home/user/PhD_Research/phd_pract_work/research_oct_nov: events
~: sudo
File Edit View Bookmarks Settings Help

```
listening on eth0, capture length 8192 bytes

1419210218.829737 warning in /nsm/bro/share/bro/base/misc/find-checksum-offloading.bro, line 5
ely receiving invalid UDP checksums, most likely from NIC checksum offloading.
```

```
user-HP-Compaq-8200-Elite-SFF-PC ~ # cd /home/user
user-HP-Compaq-8200-Elite-SFF-PC user # cd PhD_Research/phd_pract_work/research_oct_nov/
user-HP-Compaq-8200-Elite-SFF-PC research_oct_nov # gcc -g -o events event_manager_v8.c
user-HP-Compaq-8200-Elite-SFF-PC research_oct_nov # ./events
```

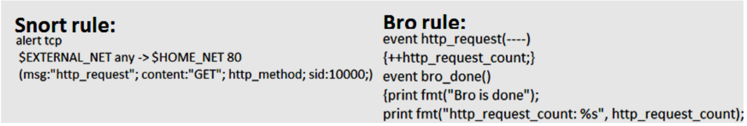
```
   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.2 IPv6 GRE (Build 78)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 1998-2011 Sourcefire, Inc., et al.
           Using libpcap version 1.1.1
           Using PCRE version: 8.12 2011-01-15
           Using ZLIB version: 1.2.3.4

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 1.15  <Build 18>
           Preprocessor Object: SF_FTPTELNET (IPV6)  Version 1.2  <Build 13>
           Preprocessor Object: SF_SIP (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS (IPV6)  Version 1.1  <Build 4>
           Preprocessor Object: SF_DCERPC2 (IPV6)  Version 1.0  <Build 3>
           Preprocessor Object: SF_REPUTATION (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP (IPV6)  Version 1.1  <Build 4>
           Preprocessor Object: SF_GTP (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP (IPV6)  Version 1.0  <Build 1>
           Preprocessor Object: SF_DNP3 (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_SMTP (IPV6)  Version 1.1  <Build 9>
           Preprocessor Object: SF_SSH (IPV6)  Version 1.1  <Build 3>
           Preprocessor Object: SF_MODBUS (IPV6)  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP (IPV6)  Version 1.0  <Build 1>
Commencing packet processing (pid=28114)
```

Figure 3: Experimental set up high attack traffic

3.2 Implementation

Bro-IDS version 2.2 and Snort version 2.9 used. In order to compare the CPU utilisation, the scripts written in two ways, to receive the HTTP traffic but do nothing, and receive the traffic and apply analysis. In Snort, rules were created and saved in myrules.rules. The main concern is to execute only this rule by commenting out all other rules and included only myrules.rules in snort.conf file. For Bro the events are created for module HTTP and saved as $PREFIX share bro site myrules.bro. All bro scripts executed from command terminal using.\bro–i–eth0 myrules.bro. Figure 4 & 5 shows a very simple script using Snort and Bro. Both the IDS programs receive traffic without any filters assigned. This configuration tests the load required just to receive and build up the data packets inside the IDS software and excludes the load required to analyse packets.

The script in Figure 5 applies a filtering process on HTTP traffic at the egress point on IDS. The HTTP analysis is programmed to apply an over-use page limit rule. This rule triggers if there is 10 HTTP requests of selected web pages within 11sec from the attacker machine to the victim. Bro scans the abnormal traffic and records results in the notice log. Snort generates an alarm file when it detects abnormal network traffic but Snort can-



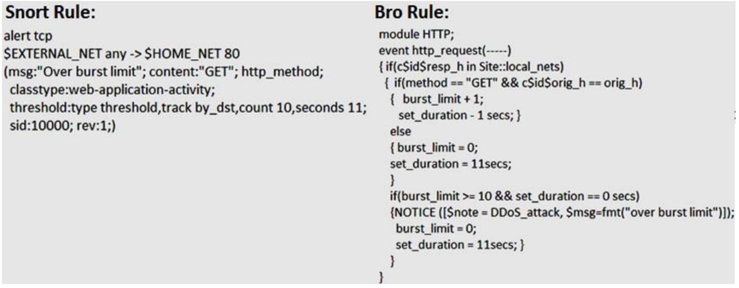
Snort rule:

```
alert tcp
 $EXTERNAL_NET any -> $HOME_NET 80
 (msg:"http_request"; content:"GET"; http_method; sid:10000;)
```

Bro rule:

```
event http_request(----)
{++http_request_count;}
event bro_done()
{print fmt("Bro is done");
print fmt("http_request_count: %s", http_request_count);
```

Figure 4: Bro and Snort programming without HTTP analysis



Snort Rule:

```
alert tcp
$EXTERNAL_NET any -> $HOME_NET 80
(msg:"Over burst limit"; content:"GET"; http_method;
 classtype:web-application-activity;
 threshold:type threshold,track by_dst,count 10,seconds 11;
 sid:10000; rev:1;)
```

Bro Rule:

```
module HTTP;
event http_request(-----)
{ if(c$id$resp_h in Site::local_nets)
  { if(method == "GET" && c$id$orig_h == orig_h)
    {  burst_limit + 1;
      set_duration - 1 secs; }
    else
    { burst_limit = 0;
    set_duration = 11secs;
    }
    if(burst_limit >= 10 && set_duration == 0 secs)
    {NOTICE ([$note = DDoS_attack, $msg=fmt("over burst limit")]);
     burst_limit = 0;
     set_duration = 11secs; }
    }
}
```

Figure 5: Bro and Snort programming with HTTP analysis

not perform complex rules as found in Bro. This work also used a complex Bro script to detect an SQL injection attack [32]. A standard Bro installation has this script in $ PREFIX\share\bro\policy\protocols\http\detect-sqli.bro. The type of attack traffic chosen was typical web page requests which are not very large. Preliminary work showed that for such traffic the load was proportional to the number of requests, the CPU load was the same for a few requests from 20 clients or the same number from one client. Attacks from clients using heavy weight payloads can be rejected based on size, similarly HTTP PUT requests can also be rejected if they are inappropriate for the page or too large.

3.3 Experimental Results

All IDS systems were tested with different attack traffic rates in four different ways to compare the CPU load;

- HTTP attack traffic and the complex rules (Bro only).
- HTTP attack traffic with HTTP analysis and rules
- HTTP attack traffic with HTTP no analysis and no rules, and
- With no load.

Each test was executed for 10 minutes, which ensured repeatable measurements with timing differences less than 0.5% between runs. Comparing Table 1 (HTTP reception with analysis and rule check) and Table 2 (HTTP reception without analysis and no rule check) for the same attack traffic rates:

Table 1: HTTP reception (analysis and rules)

Row No	Protocol	Rule	Using	DDoS attack traffic	Attack Traffic #1 (Pack/sec)	IDS CPU Load	Attack Traffic #2 (Pack/sec)	IDS CPU Load	Attack Traffic #3 (Pack/sec)	IDS CPU Load	Attack Traffic #4 (Pack/sec)	IDS CPU Load	Attack Traffic #5 (Pack/sec)	IDS CPU Load	Attack Traffic #6 (Pack/sec)	IDS CPU Load
A	HTTP	yes	BRO	yes	150	17.30%	300	21.3%	525	23.8%	3200	51.7%	4200	87.2%	6500	99.7%
B	HTTP	yes	SNORT	yes	150	2.60%	300	4.8%	525	6.8%	3200	15.3%	4200	28.7%	6500	41.3%

Table 2: HTTP reception (no analysis and no rules)

Row No	Protocol	Rule	Using	DDoS attack traffic	Attack Traffic #1 (Pack/sec)	IDS CPU Load	Attack Traffic #2 (Pack/sec)	IDS CPU Load	Attack Traffic #3 (Pack/sec)	IDS CPU Load	Attack Traffic #4 (Pack/sec)	IDS CPU Load	Attack Traffic #5 (Pack/sec)	IDS CPU Load	Attack Traffic #6 (Pack/sec)	IDS CPU Load
A	HTTP	no	BRO	yes	150	17.2%	300	19.6%	525	22.2%	3200	50.9%	4200	83.7%	6500	97.5%
B	HTTP	no	SNORT	yes	150	2.6%	300	4.7%	525	6.8%	3200	16.7%	4200	30.9%	6500	46.2%

- Note the column "Attack Traffic #2" in Table 1 and Table 2 (HTTP reception without analysis and no rule check), the CPU load of Bro increases from 19.6% to 21.3% when HTTP analysis is added. Snort has a 0.1% increase when HTTP analysis added.

- At "Attack Traffic #3", Bro shows a 1.6% increase on a base of 22.2% when HTTP analysis added. Snort shows no CPU load increase

The conclusion from this comparison is that the addition of HTTP analysis adds little to the CPU load and that most of the load is taken with no load activities plus a per received packet load. Comparing Table 1(HTTP reception and analysis) and Table 3 (no HTTP traffic) for high traffic level and zero packets/sec shows that the Bro CPU load dropped from 99.7% to 13.8% by moving from Table. 1 to Table 3.For Snort the change was 41.3% to 0.1%. Surprisingly the result shows that for the IDS software tested, the IDS CPU load increased because of the reception of traffic and not because of the processing of the traffic.

Table 4 shows the results for Bro with a complex rule for detecting SQL injection attacks. The CPU load (Table 4) is only marginally different to the other HTTP rule (Table 1) and no rules (Table 3). Again the CPU load seems related to packet reception rate and not the HTTP analysis.

3.4 IDS Load Prediction

Figure 6 & 7 graphs the packet rate against the CPU load for Snort and Bro with line of best fit. The result is a useably linear relationship between CPU load and packet rate. This means it is possible to measure the performance of a system using the low traffic configuration of one attacker and one victim as shown in Figure 2, then use regression [19, 11, 33] to produce the line of best fit and hence estimate the CPU load at higher traffic rates. This approach saves time and resources as determining CPU load at high traffic rates normally requires a powerful attack generator.

The CPU load is approximately of form given in equation number Equation (1) where a and b are constants

$$CPULoad\% = a + b * htmlpack/sec \qquad (1)$$

This results in Equation (2) which can be used to predict the CPU load at a given traffic level

$$Bro\_CPU\_load = 0.0137 * packets/sec + 15.902 \qquad (2)$$

(95% Confidence Interval = 10.19%)

$$Snort\_CPU\_load = 0.006 * packets/sec + 1.5743 \qquad (3)$$

(95% Confidence Interval = 3.91%)

Using Equation (2) for the Bro CPU load at the last data point of 6500 HTML packets/sec predicts a CPU load of 104.9% which is close to the 99.7% actually measured. Using Equation (3) for the Snort CPU load at the last data point of 6500 HTML packets/sec predicts a CPU load of 40.6% which is close to the 41.3% actually measured.

3.5 Observation

The experimental results show that the main CPU load from Bro and Snort caused by HTTP packet reception and not the analysis of that packet.

4 Power Analysis of Two-Dimensional Web Page Daemon (TDWD)

Given the surprising experimental result that most load goes into packet reception it follows that CPU load and hence power consumption would be reduced if the IDS function was built into an existing program that already performed packet reception. This may occur at the firewall or at the web server. To test this idea we developed

Table 3: Without HTTP load

Row No	Protocol	Using	DDoS attack traffic	IDS CPU Load
A	HTTP	BRO	no	13.8%
B	HTTP	SNORT	no	0.1%

Table 4: Bro with HTTP and the SQL injection rule

Row No	Protocol	Rule	Using	DDoS attack traffic	Attack Traffic #1 (Pack/sec)	IDS CPU Load	Attack Traffic #2 (Pack/sec)	IDS CPU Load	Attack Traffic #3 (Pack/sec)	IDS CPU Load
A	HTTP	SQL	BRO	yes	150	16.00%	300	18.0%	525	22.0%

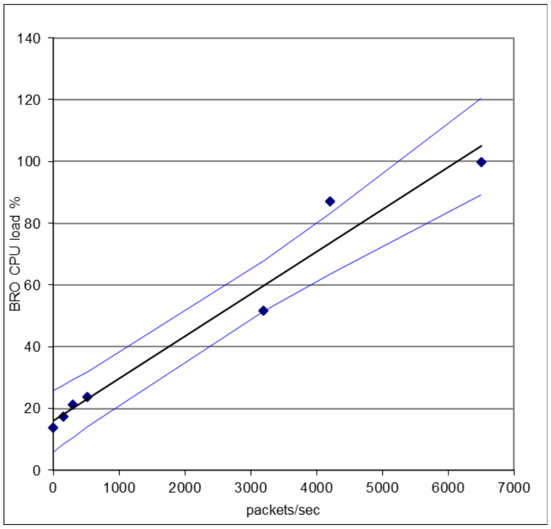


Figure 6: Bro CPU load with HTTP analysis

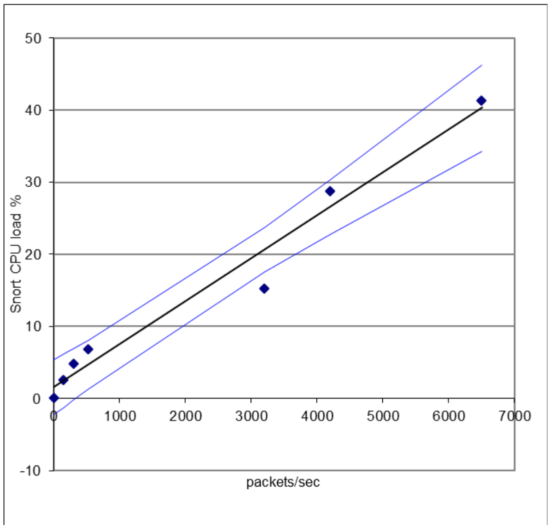


Figure 7: Snort CPU load with HTTP analysis

a novel IDS program called the Two-Dimensional Web page Daemon (TDWD) which we built into the Apache web server [27, 28]. This daemon saves web requests in a two dimensional link list, first by user IP and then by time. Analysis of this two dimensional list can determine if a user has violated a usage rule. Each web page calls a PHP script that sends the web page details to this daemon via shared memory. In these experiments the daemon then performed a rule based filter much like the Bro filter previously described that identified web page requests from the one IP at a rate of more than 10 HTTP requests in 11 seconds. The measurement and analysis of power usage was performed the new IDS daemon in the same way as done for Bro and Snort. The results are included in Table 5 in Row C. Additionally the Apache CPU load was measured at each traffic level. Row C in Table 5 and 6 shows TDWD handling all HTTP traffic and Snort and Bro given no HTTP traffic. The high traffic load of 6500 packets/second will not be analysed as the 99.7% CPU usage for Bro may mean that Bro is losing packets as suggested by the anomalous lower Apache CPU load. Using TDWD to handle HTTP saves Bro or Snort considerable CPU load and only marginally increases the CPU load on the web server as a result of running the daemon.

Comparing Table 5 (for a high traffic level of 4200 packets/sec) and Table 6(zero packets/sec) shows that the Bro CPU load dropped from approximately 87% to 14%, for Snort the change was 29% to 0.1%. This matches the scenario where all HTTP traffic is handled by TDWD. These reductions were bought at the cost of running TDWD which consumes only 3.3% of CPU load itself and approximately 10% extra CPU in Apache. The overall saving is approximately 60% CPU load for Bro and 16% for Snort. The reason for these savings is that the TDWD and Apache combination eliminates the double reception of packets and so reduces the overall CPU load on the web server system. Figure 8 & 9 graphs use data in Table 5 and Table 6 and show the CPU load of each IDS when receiving HTTP, and the resulting load on Apache.

Table 5: IDS receiving HTTP traffic with analysis and rules

Row No	Protocol	Rule	Using	DDoS attack traffic	Attack Traffic #1 (Pack/sec)	IDS CPU Load	Apache Load	Attack Traffic #2 (Pack/sec)	IDS CPU Load	Apache Load	Attack Traffic #3 (Pack/sec)	IDS CPU Load	Apache Load	Attack Traffic #4 (Pack/sec)	IDS CPU Load	Apache Load	Attack Traffic #5 (Pack/sec)	IDS CPU Load	Apache Load	Attack Traffic #6 (Pack/sec)	IDS CPU Load	Apache Load
A	HTTP	yes	BRO	yes	150	17.3%	1.7%	300	21.3%	3.6%	525	23.8%	6.3%	3200	51.7%	33.1%	4200	87.2%	39.5%	6500	99.7%	48.6%
B	HTTP	yes	SNORT	yes	150	2.6%	1.7%	300	4.8%	2.9%	525	6.8%	6.1%	3200	15.3%	30.6%	4200	28.7%	43.3%	6500	41.3%	62.9%
C	HTTP	yes	Linklist	Yes	150	0.7%	3.0%	300	1.0%	6.2%	525	1.1%	8.8%	3200	2.4%	41.7%	4200	3.3%	51.7%	6500	4.3%	67.3%

Table 6: Without HTTP load

Row No	Protocol	Using	DDoS attack traffic	IDS CPU Load
A	HTTP	BRO	no	13.8%
B	HTTP	SNORT	no	0.1%
C	HTTP	Linklist	no	0.5%

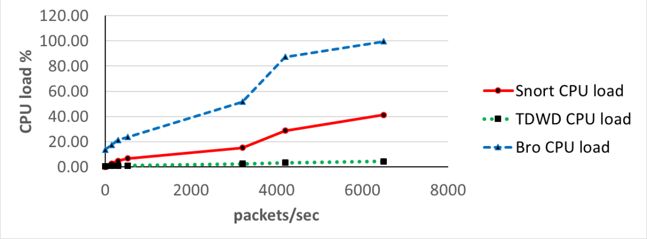


Figure 8: Bro, Snort and TDWD IDS CPU load when receiving HTTP

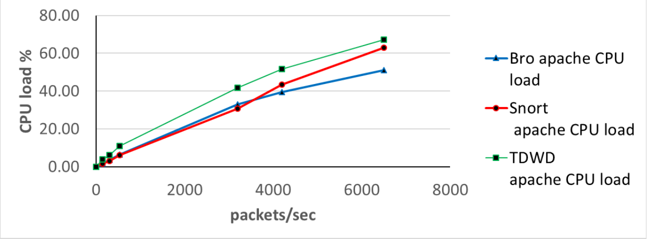


Figure 9: Bro, Snort and TDWD Apache CPU load for each IDS

4.1 Comparison of Power Usage By BRO, Snort And TDWD

The existing work described in section 2.1 showed that CPU utilisation is linearly related to power consumption. We stressed one core of an i7 CPU using the StressLinux [33] program and measured the power consumed, the results are depicted in Figure 10 with a 95% confidence interval of 0.6 watt. As per the literature there is a useably linear relationship between CPU load on one core and power consumption. Giorgio *et al.* [34] used Equation (4) to predict power consumption given CPU core utilization. Pmin is the power consumption of the entire CPU when the CPU core dedicated to the task of interest has zero utilization. Pmax is the total power consumed but the entire CPU when the task of interest is using one core at 100% utilization.

$$Power\_Consumed = (Pmax - Pmin) * Utilisation + Pmin \quad (4)$$

Given the calibration of CPU load to PC power, the power consumption of Snort, Bro, and TDWD can be graphed. From Figure 11 at high traffic rate of 6500 HTML packets/sec, Bro consumes 66 watts of the PC's power whereas Snort consumes 50 watts. The result shows that the TDWD can save approximately 25 watts compared to Bro and 10 watts compared to Snort.

This experiment has shown that an IDS that shares packet reception with another application can significantly reduce CPU use and hence power usage. The main CPU load of an IDS function is not the analysis function but a per packet load which includes the TCP stack converting between the application layer and the physical layer.

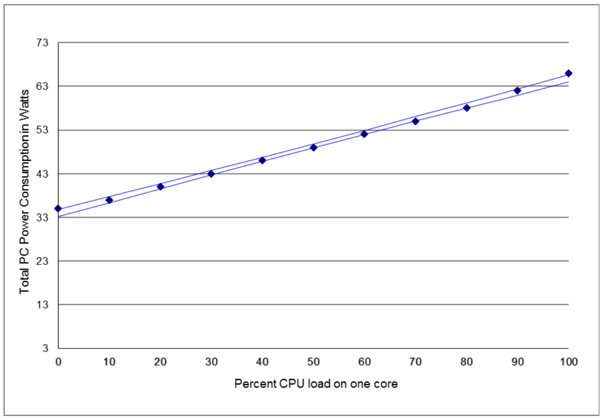


Figure 10: Power consumption vs, CPU load

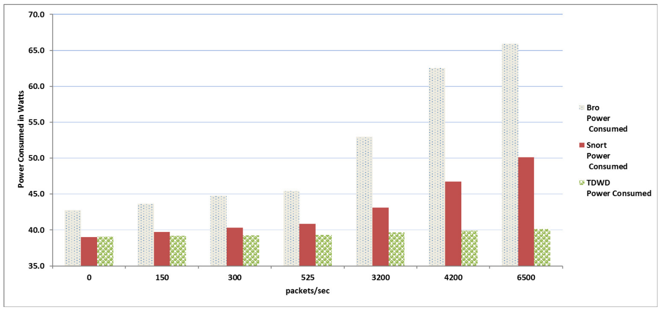


Figure 11: Power consumption of Bro, Snort and TDWD

5 Discussion of Alternative Novel Architecture

The architecture in Figure 12 shows a traditional web server with a proxy server, a firewall, a IDS and a web server where there is packet reception on each box. The same architecture is used even with a SDN implementation [40] where these functions might be applications on the one host.

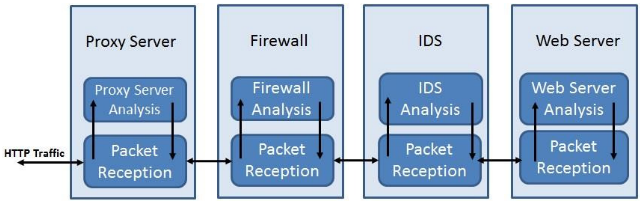


Figure 12: Traditional architecture

The research in this paper has found that the power usage in an IDS is mainly caused by packet reception rather than the packet analysis activity. This suggests that at least for SDN it would save CPU and power to have one packet reception and to pass complete IP, TCP or UDP packets between applications as shown in Figure 13.

The objection to this new architecture, for SDN or non-SDN networking, is that the security may be weaker. If any individual program is penetrated then the other programs or even the operating system may be at risk, a problem that could not occur if the functions were in separate boxes. The ability of operating systems to provide secure silos for applications is improving. One example of such an operating system is Security Enhanced- Linux (SE-Linux) [30] that provides excellent security between applications. Android has some capability in this domain as each application is treated as a separate Linux user and so the full force of the standard Linux security system is available to stop applications from interfering with each other [4]. The adoption of this new architecture is contingent on a secure operating system such as SE- Linux, being proved acceptable and having been trialled in a hostile network environment.

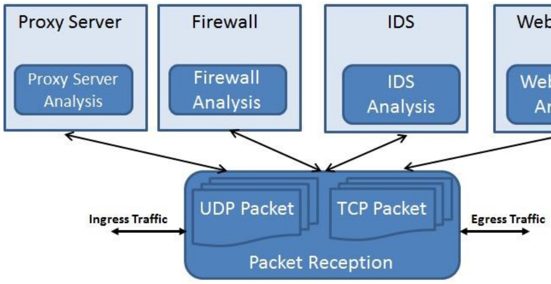


Figure 13: Novel architecture

Apache does have the internal architecture to implement the new architecture as shown in Figure 13 For example the Apache version 2 filter mod_clamav [16] scans the content delivered by the proxy module (mod_proxy) for the viruses on email using the Clamav virus scanning engine. ClamAv (Clam Antivirus) is a host based Intrusion Detection system (HIDS) written by Andreas Muller and in [25] he did mention that the processing delay has been reduced when comparing with other antivirus tools. Most likely, as this online database has shown, the removal of another packet reception process reduced CPU load. Likewise any traditional IDS like Bro or Snort could be rewritten to run as an Apache web server module and so reduced CPU load. Several modules in Apache server such as mod_status, mod_rewrite are useful for implementing server security with low CPU overhead.

6 Conclusion

A web server or web server farm may be composed of thousands of web servers and security devices. Security devices such as IDS/IPS play a crucial role in safeguarding these web servers but they do consume significant CPU time and thus electrical power. Given the large number of web server farms in the world, it is important to reduce power consumption for such farms. This paper has shown a surprising result that the power consumption of the IDS programs Snort and Bro (and most likely other security software) in a web server depends mainly the packets received per second and depends very little on the analysis performed by these programs. This makes intuitive sense as it takes significant CPU time to operate a full TCP/IP stack. Furthermore, the CPU load is a useably linear function of packets per second and so CPU load for high traffic rates can be estimated from low traffic rate measurements.

The results also have implications for the architecture of network systems. In a new proposed architecture where programs share a common CPU (as may happen with Software Defined Networks) then CPU load (and hence electrical power) can be saved if a module receives packets to the level of IP, UDP, TCP, or high level such as HTTP, and then these formed packets are shared between higher level applications. The CPU load of packet reception is done once for several applications and not repeated for every application. Security implications need careful consideration as the infection of one program may result in the easy penetration of another program or the operating system on the same CPU. Secure operating systems such as SE-Linux hold some hope of providing a secure way to implement to new architecture proposed.

The work in this paper points to several further topics for research. The first is to test whether SE-Linux or other operating systems can provide the secure software silos needed to run the new architecture. Such a system could be trialled against known attacks and then placed in a honey pot arrangement to further stress the system. Another topic worth exploring is implementing the new proposed architecture in the module based Apache web server. What security functions make sense in such an architecture where there is not a high level of security between the web server and the security programs?

References

[1] M. Akhlaq, F. Alserhani, A. Subhan, I. U. Awan, J. Mellor, and P. Mirchandani, "High speed NIDS using dynamic cluster and comparator logic," in *IEEE 10th International Conference on Computer and Information Technology (CIT'10)*, pp. 575-581, 2010.

[2] S. M. Alqahtani, M. A. Balushi, and R. John, "An intelligent intrusion detection system for cloud computing (SIDSCC'14)," in *International Conference on Computational Science and Computational Intelligence (CSCI'14)*, pp. 135-141, 2014.

[3] F. Alserhani, M. Akhlaq, I. U. Awan, J. Mellor, A. J. Cullen, and P. Mirchandani, "Evaluating intrusion detection systems in high speed networks," *Fifth International Conference on Information Assurance and Security (IAS'09)*, pp. 454-459, 2009.

[4] Android, *System Permissions*, 27 Aug. 2015. (http://developer.android.com/guide/topics/security/permissions.html)

[5] T. Bhaskar and S. D. Moitra, "A hybrid model for network security systems: Integrating intrusion detection system with survivability," *International Journal of Network Security*, vol. 7, pp. 249-260, 2008.

[6] E. Biermann, E. Cloete, and L. M. Venter, "A comparison of intrusion detection systems," *Computers & Security*, vol. 20, pp. 676–683, 2001.

[7] P. Bohrer, E. N. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, *et al.*, "The case for power management in web servers," in *Power Aware Computing*, pp. 261-289, 2002.

[8] Bro, *The Bro Network Security Monitor*, Dec. 15, 2017. (http://www.bro.org)

[9] W. Bulajoul, A. James, and M. Pannu, "Network intrusion detection systems in high-speed traffic in computer networks," *IEEE 10th International Conference on e-Business Engineering*, pp. 168-175, 2013.

[10] A. Chonka, W. Zhou, J. Singh, and Y. Xiang, "Detecting and tracing DDoS attacks by intelligent decision prototype," *Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'08)*, pp. 578-583, 2008.

[11] Colby, *LINEST in Excel*, Dec. 15, 2017. (http://www.colby.edu/chemistry/PChem/notes/linest.pdf)

[12] M. A. Eid, H. Artail, A. I. Kayssi, and A. Chehab, "LAMAIDS: A lightweight adaptive mobile agent-based intrusion detection system," *International Journal of Network Security*, vol. 6, pp. 145-157, 2008.

[13] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms," in *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'09)*, pp. 157-168, 2009.

[14] A. A. Hadi, F. H. J. Azmat, and F. H. M. Ali, "IDS using mitigation rules approach to mitigate ICMP attacks," in *International Conference on Advanced Computer Science Applications and Technologies (ACSAT'13)*, pp. 54-59, 2013.

[15] R. A. Kemmerer and G. Vigna, "Intrusion detection: A brief history and overview," *IEEE Computer*, vol. 35, no. 4, Apr. 2002.

[16] T. Kojm, *Clam AntiVirus: User Manual*, ClamAV, 2012.

[17] F. Y. Leu, J. C. Lin, M. C. Li, C. T. Yang, "A performance-based grid intrusion detection system," in *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, pp. 525-530, 2005.

[18] F. Y. Leu, J. C. Lin, M. C. Li, C. T. Yang, "Integrating grid with intrusion detection," in *19th International Conference on Advanced Information Networking and Applications*, pp. 304-309, 2005.

[19] B. Liengme, *Regression Analysis - Confidence Interval of the Line of Best Fit*, Dec. 15, 2017. (http://people.stfx.ca/bliengme/exceltips/regressionanalysisconfidence2.htm)

[20] A. X. Liu and M. G. Gouda, "Complete redundancy removal for packet classifiers in TCAMs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 424-437, 2010.

[21] P. Mehra, "A brief study and comparison of Snort and Bro open source network intrusion detection systems," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 1, pp. 383-386, 2012.

[22] D. S. A. Minaam, H. M. Abdual-Kader, and M. M. Hadhoud, "Evaluating the effects of symmetric cryptography algorithms on power consumption for different data types," *International Journal of Network Security*, vol. 11, pp. 78-87, 2010.

[23] L. Minas and B. Ellison, "The Problem of Power Consumption in Servers," *Dr. Dobb's Journal*, May 2009.

[24] T. Mukherjee, G. Varsamopoulos, S. K. S. Gupta, and S. Rungta, "Measurement-based power profiling of data center equipment," in *2007 IEEE International Conference on Cluster Computing*, pp. 476-477, 2007.

[25] A. Muller, *An Apache Virus Scanning Filter*, mod_clamav 0.23, Dec. 15, 2017. (`http://software.othello.ch/mod\_clamav/`)

[26] V. Sharma, A. Thomas, T. Abdelzaher, K. Skadron, and L. Zhijian, "Power-aware QoS management in Web servers," in *24th IEEE Real-Time Systems Symposium (RTSS'03)*, pp. 63-72, 2003.

[27] S. Sivabalan and P. J. Radcliffe, "A novel framework to detect and block DDoS attack at the application layer," in *IEEE TENCON Spring Conference*, pp. 578-582, 2013.

[28] S. Sivabalan and P. Radcliffe, "Real time calibration of DDoS blocking rules for Web Servers," *Computers*, vol. 4, pp. 42-50, 2016.

[29] N. Sklavos and P. Souras, "Economic Models and Approaches in Information Security for Computer Networks," *International Journal of Network Security*, vol. 2, pp. 14-20, 2006.

[30] S. Smalley, C. Vance, and W. Salamon, *Implementing SELinux as a Linux Security Module*, NAI Labs Report, vol. 1, pp. 139, 2001. (`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.6001\&rep=rep1\&type=pdf`)

[31] Snort, *Snort Web Page*, Dec. 15, 2017. (`http://www.Snort.org`)

[32] M. Stampar, "Inferential SQL injection attacks," *International Journal of Network Security*, vol. 18, pp. 316-325, 2016.

[33] StressLinux, *Welcome to stresslinux*, Dec. 15, 2017. (`https://www.stresslinux.org/sl/`)

[34] G. L. Valentini, S. U. Khan, and P. Bouvry, "Energy-efficient resource utilization in cloud computing," *Large Scale Network-centric Computing Systems*, John Wiley & Sons, Hoboken, NJ, USA, 2013.

[35] G. Vasiliadis, M. Polychronakis, and S. Ioannidis, "MIDeA: A multi-parallel intrusion detection architecture," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, Chicago, Illinois, USA, 2011.

[36] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability and Statistics for Engineers and Scientists*, Macmillan, New York, 1993.

[37] P. Wheeler and E. Fulp, "A taxonomy of parallel techniques for intrusion detection," in *Proceedings of the 45th Annual Southeast Regional Conference (ACM-SE'07)*, pp. 278-282, 2007.

[38] Y. Xie, S. Z. Yu, "Monitoring the application-layer DDoS attacks for popular websites," *IEEE/ACM Transactions on Networking*, vol. 17, pp. 15-25, 2009.

[39] S. Zaman and F. Karray, "Lightweight IDS based on features selection and IDS classification scheme," *International Conference on Computational Science and Engineering*, pp. 365-370, 2009.

[40] N. Zilberman, P. M. Watts, C. Rotsos, and A. W. Moore, "Reconfigurable network systems and software-defined networking," *Proceedings of the IEEE*, vol. 103, pp. 1102-1124, 2015.

Biography

Ms.Sujatha Sivabalan biography. received a B.Eng from Bharathidasan University(India)in 2004 and M.Eng from Anna University (India) in 2007.Currently doing Ph.D. in School of Electrical and Computer Engineering, RMIT University, Australia. Her research interest includes network security, DDoS attack detection and blocking.

P. J. Radcliffe biography. received a B.Eng from Melbourne University (Australia) in 1978 and worked for Ericsson Australia R&D for 7 years followed by consulting to various companies. He joined Royal Melbourne Institute of Technology (RMIT) and was awarded and M.Eng. in 1993 and a PhD in 2007. Main research interests include network protocols, Linux, and embedded systems. He received a national teaching award in 2011 and in 2012 received the RMIT early career researcher award.

A Survey to Design Privacy Preserving Protocol Using Chaos Cryptography

Hongfeng Zhu, Rui Wang
(Corresponding author: Hongfeng Zhu)

Software College, Shenyang Normal University
No. 253, HuangHe Bei Street, HuangGu District, Shenyang 110034, China
(Email:zhuhongfeng1978@163.com; 670322496@qq.com)

(Received Apr. 13, 2016; revised and accepted Sept. 25, 2016)

Abstract

Chaos theory has been widely studied and adapted in cryptography for achieving some security mechanisms, such as encryption/decryption, key agreement and hash function. The privacy of using chaos cryptography mostly relies on one of or the combination of three mechanisms: (1) Universal construction symmetric cryptography; (2) Efficient type multiplication in finite field; (3) Prudent operation XORed. This paper introduces four efficient generic methods based on three mechanisms for protecting privacy. Our four methods firstly achieve encrypted messages with mutual authentication in one-way flow. In addition, we discuss some methods about using more than two of the methods to form hybrid cases. Finally, implementation analysis, formal proof and efficiency comparison are provided to show that these mechanisms are practical, secure, and privacy preserving.

Keywords: Chaotic Maps; Privacy; Symmetric Cryptography; XORed Operation

1 Introduction

The need of mutual authentication with privacy protection is a fundamental security requirement in computer society. With wide-spread of distributed computer networks, due to most of the applications are client-server architecture, the problem of only legal users have access to use the various remote services has attracted much attention. (see [9, 10, 23, 25]). Combined with the recent trend, chaos theory has widely used to cryptography. Chaotic system has numerous advantages, such as extremely sensitive to initial parameters, unpredictability, boundness, etc. Meanwhile, chaotic sequence generated by chaotic system has the properties of non-periodicity and pseudorandomness.

In 1998, Baptista [1] firstly connects cryptography with chaos theory. As a fundamental cryptographic primitive, key agreement protocol allows two or more parties to agree on shared keys which will be used to protect their later communication. Then, combining chaos theory and key agreement primitive, many authenticated key exchange (AKE) protocols [19] have been proposed. The literature [34] firstly proposed a new one-way authenticated key agreement scheme (OWAKE) based on chaotic maps with multi-server architecture [21, 36]. The OWAKE scheme is widely used to no need for mutual authentication environment on Internet, such as readers-to-journalists model and patient-to-expert model. Using the chaotic maps, the literature [32] firstly proposed a new multiple servers to server architecture (MSTSA) to solve the problems caused by centralized architecture, such as multi-server architecture with the registration center (RC). The core ideas of the proposed scheme are the symmetry (or called peer to peer) in the servers side and the transparency for the clients side. In brief, based on chaotic maps, there were many AKE protocols from functionality aspect, or from efficiency aspect, or from security aspect, or from architecture aspect to improve the AKE protocols. For capturing more functionality, identity-based MSAKA protocols, based on bilinear pairings or elliptic curve cryptosystem (ECC) MSAKA protocols, dynamic identity-based MSAKA protocols and other MSAKA protocols came up recently [6, 7, 19, 27].

Recently, many schemes based on chaos theory are proposed [8, 16, 24, 26, 27, 35]. Compared with the related other schemes, these schemes avoid numerous complex operations. One direction is about static/dynamic Identity (ID) authentication schemes [24, 26] which are based on chaotic maps. But the literature [35] pointed out that Lin's scheme [24] cannot resist dictionary attack, user spoofing attack, denial of service attack and exclusive-or operation with pad operation leaking attack. In 2013, Guo *et al.* [8] proposed a chaotic maps-based key agreement protocol which avoided modular exponential computing and scalar multiplication on elliptic curve. Nowadays, with the fast development of Internet, privacy protection of users is a hot issue. In 2014, Liu *et al.* [16] proposed a multi-function password mutual authentication key agreement scheme with privacy preserving. In 2015, Zhu *et al.* [29, 33] proposed an even more efficient scheme which is only used chaotic maps for mutual authentication instead of encrypting/decrypting messages transferred between user and server, and the users' privacy information is also protected.

There are so many AKE, dynamic ID and others' schemes using chaotic maps so that we cannot introduction by one. In our opinions, these schemes can be refined and further to form some methods or primitive units of protocol. Our goals are to sum up the core contents to serve for constructing diversified schemes with chaotic maps rather than designing a concrete and nonadaptive scheme.

In this paper, we sum up four methods to capturing privacy attribute based on chaotic maps and some expandable forms which can construct many security protocol with privacy preserving. The main contributions of this paper are shown below:

- In Symmetric Encryption Method, we propose a method to design a privacy preserving protocol with mutual authentication in one-way flow using chaotic maps, secure symmetric encryption/decryption and a secure hash function. This kind of method will provide all sensitive information, such as identity, timestamp, value of hash and so on. That will give the attacker the maximum limit of attacks.

- Multiplication in Finite Field method. For achieving both efficiency and privacy preserving attribute, we only use chaotic maps and secure hash function and eliminate secure symmetric encryption/decryption. Our new idea is first to construct a ciphertext based on opposite side's public key and own chosen random number, and then combine both side's public keys with one-time hash value to compute an authenticator for achieving mutual authentication and one-way flow.

- XORed method. For improving efficiency further, we adopt XORed operation instead of Multiplication in Finite Field. This method must pay attention to leak any bits. Aimming at this method, we sum up two rules to resist the potential risk: make the same length about the two sides of $\oplus$ and keep strict secret for the two sides of $\oplus$.

- We sum up many hybrid modes which can design many new protocols with privacy preserving for adapting to changeable environments (see Section 4). We also sum up the security proofs methods and give many literatures to refer (see Section 5).

The rest of the paper is organized as follows: Some preliminaries are given in Section 2. In Section 3, we describe three technologies for privacy with chaotic maps. In Section 4, we discuss some evolved methods. The efficiency analysis of our proposed protocol and methods of provable security are given in Section 5. This paper is finally concluded in Section 6.

2 Chebyshev Chaotic Maps

Let n be an integer and let x be a variable with the interval $[-1, 1]$. The Chebyshev polynomial [24] $T_n(x) : [-1, 1] \to [-1, 1]$ is defined as $T_n(x) = \cos(ncos^{-1}(x))$. Chebyshev polynomial map $T_n : R \to R$ of degree n is defined using the following recurrent relation:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x),$$

$$\text{where } n \geq 2, T_0(x) = 1, \text{ and } T_1(x) = x.$$

The first few Chebyshev polynomials are:

$$T_2(x) = 2x^2 - 1, T_3(x) = 4x^3 - 3x, T_4(x) = 8x^4 - 8x^2 + 1, ...$$

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that

$$T_r(T_s(x)) = T_{rs}(x).$$

An immediate consequence of this property is that Chebyshev polynomials commute under composition

$$T_r(T_s(x)) = T_s(T_r(x)).$$

Because it is actually proven insecure in literature [2] that Chebyshev polynomials are running the polynomial on decimal number, we adopts the enhanced Chebyshev polynomials to design our protocols. In order to enhance the security, Zhang [28] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$.

Definition 1. *(Enhanced Chebyshev polynomials)The enhanced Chebyshev maps of degree $n(n \in N)$ are defined as: $T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x))(\text{mod} p)$, where $n \geq 2, x \in (-\infty, +\infty)$, and p is a large prime number. Obviously, $T_{rs}(x) = T_r(T_s(x)) = T_s(T_r(x))$.*

Definition 2. *(DLP, Discrete Logarithm Problem)Given an integer a,find the integer r, such that $T_r(x) = a$.*

Definition 3. *(CDH, Computational* Diffie − −Hellman *Problem)Given an integer x, and the values of $T_r(x), T_s(x)$, what is the value of $T_{rs}(x) =?$.*

It is widely believed that there is no polynomial time algorithm to solve DLP, CDH with a non-negligible probability.

3 Three Technologies for Privacy with Chaotic Maps

In this section, we present three general technologies for privacy with chaotic maps, including the methods, extended methods and some deductive ways. Simply speaking, for all the nodes $node_i (1 \leq i \leq n)$, their public keys are $(x, T_{K_i}(x))(1 \leq i \leq n)$ and the corresponding secret keys are $K_i (1 \leq i \leq n)$. And without loss of generality, we assume a user or a server $node_1$ is the sender, and the others $node_i (2 \leq i \leq n)$ are the receivers. Due to space limitation in this paper, we are not able to discuss the details about how to distribute the public-private key pairs of the users. Some notations hereafter are shown in Table 1.

Table 1: Notations

Symbol	Definition
ID_i	The identity of nodes
a, b, r_i	Nonces
$(x, T_{K_i}(x))$	Public key of $node_i$ based on Chebyshev chaotic maps
K_i	Secret key of $user_i$ based on Chebyshev chaotic maps
$E_K()/D_K()$	A pair of secure symmetric encryption/decryption functions with the key K
M, H, T, F	Plaintext, a secure chaotic maps-based hash function, Timestamp and Pseudo-random function
$\|\|$	Concatenation operation
...	Some other information

3.1 Privacy Protection Mechanism I: Symmetric Encryption

The method of Privacy Protection Mechanism I is shown in Figure 1.

Step 1. When $node_1$ wants to send an encrypted messages with mutual authenticator in one-way flow, it chooses a random a and the public key of the peer to compute:$T_a(x)$, $K_{1i} = T_a T_{K_i}(x)$, $H_i(M_i||T||...)$, $C_i = E_{K_{1i}}(M_i||H_i||T||T_{K_1}T_{K_i}(x)...)$. Then $node_1$ sends $\{T_a(x), C_i\}$ to the others peers.

Step 2. Upon receiving $\{T_a(x), C_i\}$ from the $node_1$, the $node_i$ computes $K_{1i} = T_{K_i}T_a(x)$ and uses K_{1i} to decrypt C_i. Then, $node_i$ can get the $M_i||H_i||T||T_{K_1}T_{K_i}(x)...$ and do the following tasks:

1) To verify the timestamp T;

2) If timestamp authentication passed, $node_i$ computes hash value and compare it with the H_i;

3) If the hash value authentication passed, $node_i$ computes $T_{K_i}T_{K_1}(x)$ and compare with $T_{K_1}T_{K_i}(x)$. If they are equals, that means $node_1$ is the real $node_1$.

Finally, $node_i$ will accept the messages in this flow.

Remark 1. *1) The timestamp T is encrypted during all the communication process which can resist the common interrupt attack. If the timestamp T is plaintext on the public channel, the adversary only makes the timestamp smaller simply so that let the other peer authentication fail.*

2) The information $T_{K_1}T_{K_i}(x)$ is the authenticator which can let the receiver authenticate the sender while no need for another exchange, in other words, mutual authentication can be achieved in one communication which is an efficient method. Although the $T_{K_1}T_{K_i}(x)$ is invariant, it is encrypted and the C_i is always changing. So our Privacy Protection Mechanism I is efficient and secure.

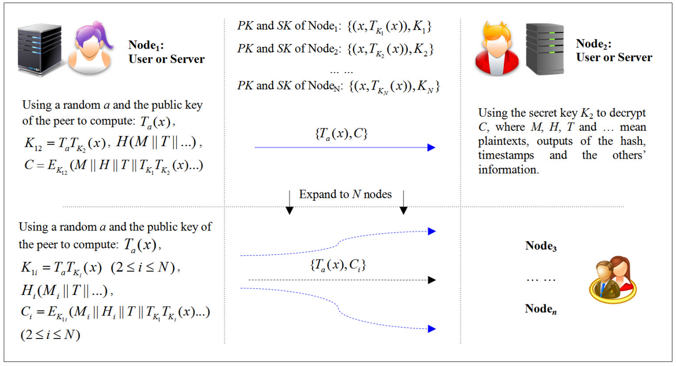

Node$_1$:
User or Server
PK and *SK* of Node$_1$: $\{(x, T_{K_1}(x)), K_1\}$
PK and *SK* of Node$_2$: $\{(x, T_{K_2}(x)), K_2\}$
... ...
PK and *SK* of Node$_N$: $\{(x, T_{K_N}(x)), K_N\}$
Node$_2$:
User or Server
Using a random a and the public key of the peer to compute: $T_a(x)$,
$K_{12} = T_a T_{K_2}(x)$, $H(M \| T \| ...)$,
$C = E_{K_{12}}(M \| H \| T \| T_{K_1} T_{K_2}(x)...)$
$\{T_a(x), C\}$
Using the secret key K_2 to decrypt C, where M, H, T and ... mean plaintexts, outputs of the hash, timestamps and the others' information.
Expand to N nodes
Using a random a and the public key of the peer to compute: $T_a(x)$,
$K_{1i} = T_a T_{K_i}(x) \ (2 \leq i \leq N)$,
$H_i(M_i \| T \| ...)$,
$C_i = E_{K_{1i}}(M_i \| H_i \| T \| T_{K_1} T_{K_i}(x)...)$
$(2 \leq i \leq N)$
$\{T_a(x), C_i\}$
Node$_3$
... ...
Node$_n$

Figure 1: The method of privacy protection mechanism I: Symmetric encryption

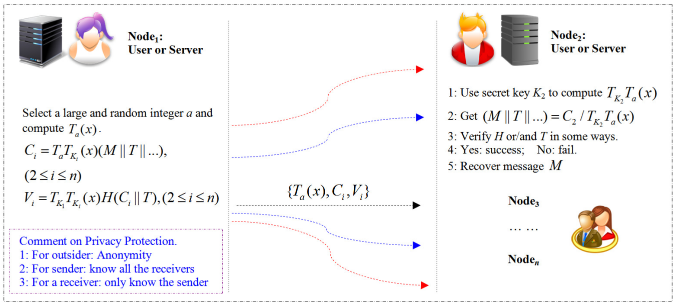

Node$_1$:
User or Server
Node$_2$:
User or Server
Select a large and random integer a and compute $T_a(x)$.
$C_i = T_a T_{K_i}(x)(M \| T \| ...)$,
$(2 \leq i \leq n)$
$V_i = T_{K_1} T_{K_i}(x) H(C_i \| T), (2 \leq i \leq n)$
Comment on Privacy Protection.
1: For outsider: Anonymity
2: For sender: know all the receivers
3: For a receiver: only know the sender
$\{T_a(x), C_i, V_i\}$
1: Use secret key K_2 to compute $T_{K_2} T_a(x)$
2: Get $(M \| T \| ...) = C_2 / T_{K_2} T_a(x)$
3: Verify H or/and T in some ways.
4: Yes: success; No: fail.
5: Recover message M
Node$_3$
... ...
Node$_n$

Figure 2: The 1-to-N method of privacy protection mechanism II: Multiplication in finite field

3.2 Privacy Protection Mechanism II: Multiplication in Finite Field

The 1-to-N method of Privacy Protection Mechanism II is shown in Figure 2.

Step 1. When $node_1$ wants to send an encrypted messages with mutual authenticator in one-way flow, it chooses a random a and the public key of the peer to compute:$T_a(x)$, $C_i = T_a T_{K_i}(x)(M||T||...)$, $V_i = T_{K_1}T_{K_i}(x)H(C_i||T), (2 \leq i \leq n)$. Then $node_1$ sends $\{T_a(x), C_i, V_i\}$ to the others peers.

Step 2. Upon receiving $\{T_a(x), C_i, V_i\}$ from the $node_1$, the $node_i$ computes $T_{K_i}T_a(x)$ and uses it to decrypt C_i. Then, $node_i$ can get the $(M||T||...) = C_2/T_{K_2}T_a(x)$ and do the following tasks:

1) To verify the timestamp T;
2) If timestamp authentication passed, $node_i$ computes hash value $V_i^{'} = T_{K_i}T_{K_1}(x)H(C_i||T)$ and compare it with the V_i;
3) If the authentication passed, that means $node_1$ is the real $node_1$.

Finally, $node_i$ will accept the messages in this flow.

Remark 2:

1) The timestamp T is encrypted during all the communication process which can resist the common interrupt attack. If the timestamp T is plaintext on the public channel, the adversary only makes the timestamp smaller simply so that let the other peer authentication fail.
2) The authenticator $V_i = T_{K_1}T_{K_i}(x)H(C_i||T)$ is always changing because T and C_i are different for each interaction and each node. So the 1-to-N Method of Privacy Protection Mechanism II can also achieve mutual authentication in one communication.

Step 1. When the $node_1$ wants to send the same message m to the $node_i (2 \leq i \leq n)$, it chooses two large and random integers a and b. Next, the $node_1$ computes $T_a(x)$,$T_b(x)$,$C_i = T_bT_{K_i}(x)ID_1, (2 \leq i \leq n)$, $V_i = T_a(x)T_{K_1}T_{K_i}(x), (2 \leq i \leq n)$, $W = T_a(x)m$ and $F_i = F_{T_a(x)}(C_i||V_i||W), (2 \leq i \leq n)$. Finally, U_1 sends $\{T_b(x), C_i, V_i, W, F_i\}$ to the users $U_i (2 \leq i \leq n)$.

Step 2.

1) Upon receiving $\{T_b(x), C_i, V_i, W, F_i\}$ from the sender, firstly, any node can recover the identity of the sender by using secret key K_i to compute $T_{K_i}T_b(x)$ and get $ID_1 = C_i/T_{K_i}T_b(x)$.
2) Based the sender's identity ID_1, $node_i$ can get the public key $T_{K_1}(x)$ and compute $T_{K_i}T_{K_1}(x)$ for getting $T_a(x) = V_i/T_{K_i}T_{K_1}(x)$. This step is also authenticating the sender, if the sender is the sender In the last step, any user can recover the right message, if not, the recovered message will not be the plaintext.
3) U_i authenticates the message integrity $F_{T_a(x)}(C_2||V_2||W) = F_2?$. If yes, the ciphertext is valid. Otherwise, the ciphertext is invalid or has been damaged during transmission.
4) Finally, based on their secret key Ki, any node in the group can recover the message $m = \frac{W}{V_i/T_{K_i}T_{K_1}(x)}$.

Remark 3: In this method, we use pseudo-random function instead of hash function for achieving in the standard model. You can also use hash function for getting high-efficiency in the random oracle.

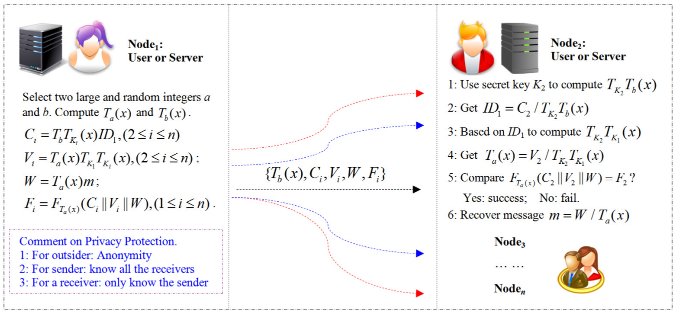

Node$_1$:
User or Server
Select two large and random integers a and b. Compute $T_a(x)$ and $T_b(x)$.
$C_i = T_bT_{K_i}(x)ID_1, (2 \leq i \leq n)$
$V_i = T_a(x)T_{K_1}T_{K_i}(x), (2 \leq i \leq n)$;
$W = T_a(x)m$;
$F_i = F_{T_a(x)}(C_i \| V_i \| W), (1 \leq i \leq n)$.
Comment on Privacy Protection.
1: For outsider: Anonymity
2: For sender: know all the receivers
3: For a receiver: only know the sender
$\{T_b(x), C_i, V_i, W, F_i\}$
Node$_2$:
User or Server
1: Use secret key K_2 to compute $T_{K_2}T_b(x)$
2: Get $ID_1 = C_2 / T_{K_2}T_b(x)$
3: Based on ID_1 to compute $T_{K_2}T_{K_1}(x)$
4: Get $T_a(x) = V_2 / T_{K_2}T_{K_1}(x)$
5: Compare $F_{T_a(x)}(C_2 \| V_2 \| W) = F_2$?
Yes: success; No: fail.
6: Recover message $m = W / T_a(x)$
Node$_3$
… …
Node$_n$

Figure 3: The chained method of privacy protection mechanism II: Multiplication in finite field

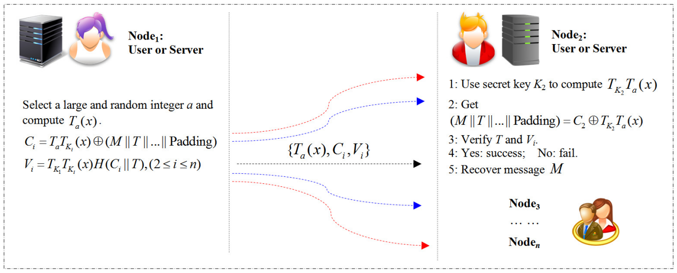

Node$_1$:
User or Server
Select a large and random integer a and compute $T_a(x)$.
$C_i = T_aT_{K_i}(x) \oplus (M \| T \| ... \| \text{Padding})$
$V_i = T_{K_1}T_{K_i}(x)H(C_i \| T), (2 \leq i \leq n)$
$\{T_a(x), C_i, V_i\}$
Node$_2$:
User or Server
1: Use secret key K_2 to compute $T_{K_2}T_a(x)$
2: Get
$(M \| T \| ... \| \text{Padding}) = C_2 \oplus T_{K_2}T_a(x)$
3: Verify T and V_i.
4: Yes: success; No: fail.
5: Recover message M
Node$_3$
… …
Node$_n$

Figure 4: The method of privacy protection mechanism III: XORed operation

3.3 Privacy Protection Mechanism III: XORed Operation

The method of Privacy Protection Mechanism III is shown in Figure 4.

Step 1. When $node_1$ wants to send an encrypted messages with mutual authenticator in one-way flow, it chooses a random a and the public key of the peer to compute:$T_a(x)$, $C_i = T_aT_{K_i}(x) \oplus (M||T||...||\text{Padding})$,$V_i = T_{K_1}T_{K_i}(x)H(C_i||T), (2 \leq i \leq n)$. Then $node_1$ sends $\{T_a(x), C_i, V_i\}$ to the others peers.

Step 2. Upon receiving $\{T_a(x), C_i, V_i\}$ from the $node_1$, the $node_i$ computes $T_{K_i}T_a(x)$ and uses it to decrypt C_i. Then, $node_i$ can get the $(M||T||...||\text{Padding}) = C_2 \oplus T_{K_2}T_a(x)$ and do the following tasks:

1) To verify the timestamp T;
2) If timestamp authentication passed, $node_i$ computes hash value $V_i^{'} = T_{K_i}T_{K_1}(x)H(C_i||T)$ and compare it with the V_i;
3) If the authentication passed, that means node1 is the real $node_1$.

Finally, $node_i$ will accept the messages in this flow.

Remark 4:

1) In this method, XORed operation may be lead some potential risk. The messages $(T_aT_{K_i}(x)$

and $(M||T||...||\text{Padding})$) of both sides for the XORed operation cannot leak any bits. Otherwise, the $T_aT_{K_i}(x)$ may leak the same bits. For example, in $C_i = T_aT_{K_i}(x) \oplus (M||T||...||\text{Padding})$, the M may include the identity of a node. An adversary get the identity in some way, so the correspondent bits of the $T_aT_{K_i}(x)$ will be leak. The concrete details can be found in [12].

2) In some way, the padding may also leak some bits of the end of the $T_aT_{K_i}(x)$.Because the padding mode is usually public. The best method to make Privacy Protection Mechanism III secure are no padding (cut out part of $T_aT_{K_i}(x)$ to make the same length with $(M||T||...)$), and at the same time don't leak any message of the $(M||T||...)$.

4 The Discussions about Hybrid or Evolved Schemes

For simplicity, we make four methods (Symmetric Encryption, 1-to-N Method of Multiplication in Finite Field, Chained Method of Multiplication in Finite Field and XORed Operation) expressed as Algorithm 1, Algorithm 2, Algorithm 3, Algorithm 4.

4.1 Composable Mode

The four methods can be combined to achieve diversified security protocol which may be more efficient or more functions. We set some examples as follows:

Algorithm 1 + Algorithm 2: $C_i = E_{T_aT_{K_i}(x)}(M_i||T...)$, $V_i = T_{K_1}T_{K_i}(x)H(C_i||T)$, $(2 \leq i \leq n)$. This is a kind of composable mode uses Symmetric Encryption to protect messages and privacy and adopts multiplication to provide the mutual authentication.

Algorithm 1 + Algorithm 4: $C_i = E_{T_aT_{K_i}(x)}(M_i||T...) \oplus T_{K_1}T_{K_i}(x), V_i = H(C_i||T), (2 \leq i \leq n)$. This is a kind of composable mode can improve the security level, and the authenticator is a simple value of hash function.

4.2 Modified Mode

The four methods can be modified to diversified security protocol which may be more efficient or more functions. We set some examples as follows:

1) Modified the authenticator: For Algorithm 4, we can make the authenticator $V_i = T_{K_1}T_{K_i}(x)H(C_i||T)$ become $V_i = H(T_{K_1}T_{K_i}(x)) \oplus H(C_i||T)$. This modification can get more efficient. Because the modified edition uses one XORed and one hash function instead of one multiplication, which is more efficient than before (see Section 5.1).

2) Modified the communication round number: For Algorithm 1, Algorithm 2, Algorithm 4, you can use multiple communication to improve computational efficiency and eliminate the $T_{K_1}T_{K_i}(x)$ or others information.

3) For Algorithm 3, the encrypted message $W = T_a(x)m$ can be modified into $W = T_a(x) \oplus m$. This will be more efficient because compared with Multiplication, XORed operation can be ignored. Both sides of XORed, $T_a(x)$ and m cannot be leak any bits (see Remark 4).

4.3 Extended Mode (Three Nodes and Key Agreement)

The four methods can be extended to three-party or N-party schemes. We can use Algorithm 1 or Algorithm 2 or Algorithm 4 repeatedly can achieve three-party key exchange/key distribution or encrypted messages with mutual authentication.

About N-party group key agreement schemes, we can use Algorithm 1 or Algorithm 2 or Algorithm 4 repeatedly and refer the literature [30] to divide N-party schemes into two phases: the first phase is two-party exchange phase, the second phase is group key generated phase. From Table 2, we can see the general structure about N-party group key agreement schemes based on Algorithm 1. And the examples for Algorithm 2 or Algorithm 4 we just omit for simplicity.

4.4 Functional Mode (Multiple Keys Agreement)

In this mode, we only provide multiple Keys agreement for two-party. Just like the literature [31], we can also use Algorithm 1 or Algorithm 2 or Algorithm 4 repeatedly to achieve two-party get multiple keys in an agreement instance. The multiple keys scheme can be divided into two phases: the first phase is authenticated and transmit secret shadows phase, the second phase is Non-interactive key establishment with privacy preserving phase. From Table 3, we can see the general structure about multiple keys agreement schemes based on Algorithm 2. And the examples for Algorithm 1 or Algorithm 4 we just omit for simplicity.

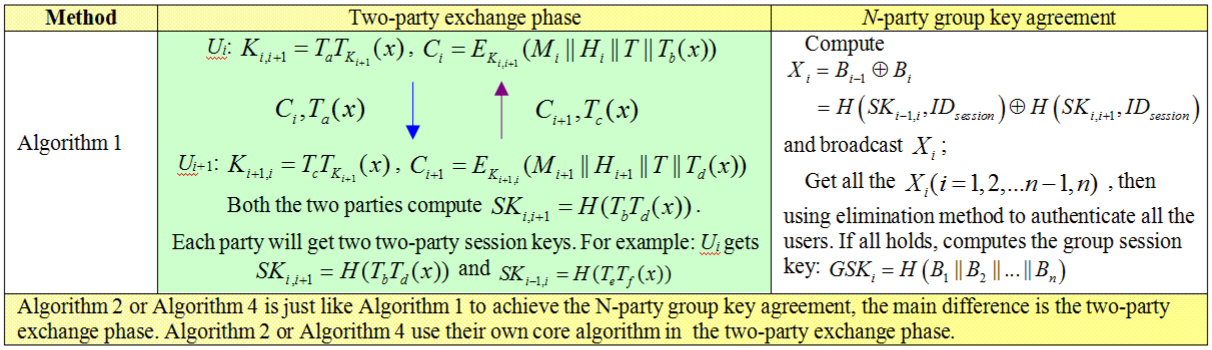
In brief, we can design many hybrid or evolved schemes from our four basic methods. We can't list all the schemes, even the listed schemes, we just describe the main structure and omit the details.

5 Features Comparison

5.1 The Sum up about Our Four Methods Efficiency

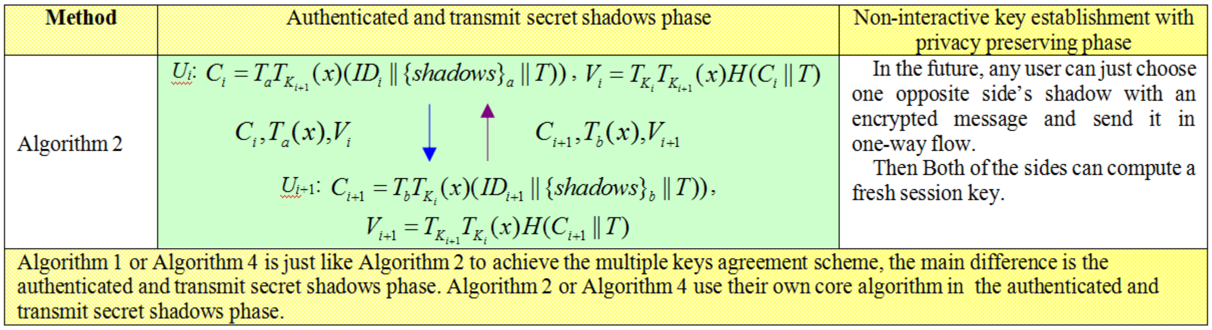
Because the paper given the methods which are the universal formulations can be used directly by any specific

Table 2: N-party group key agreement scheme based on Algorithm 1



Method	Two-party exchange phase	*N*-party group key agreement
Algorithm 1	U_i: $K_{i,i+1} = T_a T_{K_{i+1}}(x)$, $C_i = E_{K_{i,i+1}}(M_i \parallel H_i \parallel T \parallel T_b(x))$ $C_i, T_a(x)$ ↓ ↑ $C_{i+1}, T_c(x)$ U_{i+1}: $K_{i+1,i} = T_c T_{K_{i+1}}(x)$, $C_{i+1} = E_{K_{i+1,i}}(M_{i+1} \parallel H_{i+1} \parallel T \parallel T_d(x))$ Both the two parties compute $SK_{i,i+1} = H(T_b T_d(x))$. Each party will get two two-party session keys. For example: U_i gets $SK_{i,i+1} = H(T_b T_d(x))$ and $SK_{i-1,i} = H(T_e T_f(x))$	Compute $X_i = B_{i-1} \oplus B_i$ $= H(SK_{i-1,i}, ID_{session}) \oplus H(SK_{i,i+1}, ID_{session})$ and broadcast X_i; Get all the $X_i (i = 1, 2, \ldots n-1, n)$, then using elimination method to authenticate all the users. If all holds, computes the group session key: $GSK_i = H(B_1 \parallel B_2 \parallel \ldots \parallel B_n)$
Algorithm 2 or Algorithm 4 is just like Algorithm 1 to achieve the N-party group key agreement, the main difference is the two-party exchange phase. Algorithm 2 or Algorithm 4 use their own core algorithm in the two-party exchange phase.		

Table 3: Multiple keys agreement scheme based on Algorithm 2



Method	Authenticated and transmit secret shadows phase	Non-interactive key establishment with privacy preserving phase
Algorithm 2	U_i: $C_i = T_a T_{K_{i+1}}(x)(ID_i \parallel \{shadows\}_a \parallel T))$, $V_i = T_{K_i} T_{K_{i+1}}(x) H(C_i \parallel T)$ $C_i, T_a(x), V_i$ ↓ ↑ $C_{i+1}, T_b(x), V_{i+1}$ U_{i+1}: $C_{i+1} = T_b T_{K_i}(x)(ID_{i+1} \parallel \{shadows\}_b \parallel T))$, $V_{i+1} = T_{K_{i+1}} T_{K_i}(x) H(C_{i+1} \parallel T)$	In the future, any user can just choose one opposite side's shadow with an encrypted message and send it in one-way flow. Then Both of the sides can compute a fresh session key.
Algorithm 1 or Algorithm 4 is just like Algorithm 2 to achieve the multiple keys agreement scheme, the main difference is the authenticated and transmit secret shadows phase. Algorithm 2 or Algorithm 4 use their own core algorithm in the authenticated and transmit secret shadows phase.		

scheme, not the concrete schemes, we can not give any comparisons about efficiency with some related literatures. From Table 4, we can conclude that our four methods have high-efficient property.

All the computational time of some common algorithms can be summarized as follows [11, 12]:

T_{Mac}: The time for executing a strongly unforgeable MAC algorithm computation;

T_F: The time for executing a secure pseudorandom function computation$T_F \approx 4T_{MUL}$;

$T_{MUL/EXP/INV}$: The time for computing a modular multiplication/exponentiation/inversion $T_{EXP} \approx 240T_{MUL}$, $T_{INV} \approx 10T_{MUL}$);

T_H: The time for computing a one-way hash function computation ($T_H \approx 4T_{MUL}$);

$T_{EM/EA}$: The time for computing a point multiplication/addition operation over an elliptic curve($T_{EM} \approx 29T_{MUL}$, $T_{EA} \approx 0.12T_{MUL}$);

$T_{SE/SD}$: The time for performing a symmetric encryption/decryption algorithm computation ($T_{SE} \approx T_H \approx 4T_{MUL}$, $T_{SD} \approx T_H \approx 4T_{MUL}$);

T_C: The time for executing enhanced Chebyshev polynomial ($T_C \approx 60T_H$); T_{XOR}: The computational cost of XOR operation could be ignored when compared with other operations.

Based on the Table 4, we can conclude that the efficiency ranking is: $\mathbf{III > II : ChainedMethod >> II : 1-to-NMethod > I}$.

5.2 The Sum up about Our Four Methods Security

For simplicity, we make four methods (Symmetric Encryption, 1-to-N Method of Multiplication in Finite Field, Chained Method of Multiplication in Finite Field and XORed Operation) expressed as Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4.

1) The standard model: This model must pay attention to not use hash function. The Algorithm 1 can be proved just like the literature [17]. The Algorithm 2 and Algorithm 3 can be proved just like the literature [14]. The literature [14] is also as the reference for Algorithm 4, but what calls for special attention is that the both sides for the XORed operation must not leak any bit information.

2) The random oracle model: This model can use hash function for achieving high efficiency. In this model, you can firstly define some roles, such as the users, the servers and the adversary. Next, this model should let the adversary make some following oracle queries, such as send, reveal, corrupt, test and so on. Finally, this model should define some secure goals,

Table 4: Our proposed scheme efficiency

Privacy Protection Method	Main features	Party	Efficiency(for one node)
I: Symmetric Encryption	Only one-way Flow can achieve Encrypted message Delivered with mutual Authentication	Sender	$(n-1)(T_H+2T_C+T_{SE/SD}) \approx (n-1)488T_{MUL}$
		Receivers(n-1)	$T_H+2T_C+T_{SE/SD} \approx 488T_{MUL}$
II: Multiplication in Finite Field 1-to-N Method		Sender	$(n-1)(T_H+2T_C+2T_{MUL}) \approx (n-1)486T_{MUL}$
		Receivers(n-1)	$T_H+2T_C+2T_{MUL} \approx 486T_{MUL}$
II: Multiplication in Finite FieldChained Method		Sender	$nT_F+2nT_C+(2n+1)T_{MUL} \approx (126n+1)T_{MUL}$
		Receivers(n-1)	$T_F+2T_C+3T_{MUL} \approx 127T_{MUL}$
III: XORed Operation		Sender	$(n-1)(T_H+2T_C+T_{MUL}+T_{XOR}) \approx (n-1)125T_{MUL}$
		Receivers(n-1)	$T_H+2T_C+T_{MUL}+T_{XOR} \approx 125T_{MUL}$

T_H: Time for Hash operation.
T_F: Time for Pseudo-random function.
$T_{SE/SD}$: Time for Symmetric parametric function.
T_{MUL}: Time for Integer multiplication operation in the field.
T_{XOR}: Time for XOR operation.The computational cost of XOR operation could be ignored.
T_C: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial using the algorithm in literature [18].

for example, matching conversations, secure mutual authentication and secure key exchange. The literature [3, 4] can be as a reference for Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4.

3) The logical analysis method: BAN-like logics [5] is one of the main tools for analysis cryptographic protocols in recent years, the limitations of BAN logic are analyzed and illustrated with examples, and then the features of the extended BAN-like logics and their common defects are studied.

6 Conclusion

In this paper, we propose MRCM, a novel scheme towards building a PKC-based scheme for a sender sending only one encrypted message with some authentication information to multi-receiver, and at the same time, achieving privacy protection. The core idea we have followed is that the most existing multi-receiver schemes are bilinear pairing-based, for improving the efficiency, should be exploited to securely change another efficient cryptosystem, such as, chaotic maps in this paper. Since the hash function is not used, and chaotic maps is adopted to a new encrypted algorithm without using symmetrical encryption, the proposed solution offers significant advantages (the standard model and high-efficiency) with respect to a traditional multi-receiver protocols. Compared with the related works, our MRCM scheme is not the trade off between security and efficiency, but is comprehensively improved scheme.

Acknowledgments

This work is supported by the Liaoning Provincial Natural Science Foundation of China (Grant No. 201602680).

References

[1] M. S. Baptista, ”Cryptography with chaos,” *Physics Letters A*, vol. 240, no. 1, pp. 50–54, 1998.

[2] P. Bergamo, P. DArco, A. D. Santis, L. Kocarev, ”Security of public-key cryptosystems based on chebyshev polynomials,” *IEEE Transactions on Circuits and Systems. Part I: Regular Papers*, vol. 52, no. 7, pp. 1382–1393, 2005.

[3] M. Bellare, P. Rogaway, ”Entity authentication and key distribution,” in *Advances in Cryptology (CRYPTO'93)*, pp. 232–249, 1993.

[4] M. Bellare, D. Pointcheval, P. Rogaway, ”Authenticated key exchange secure against dictionary attacks,” in *Advances in Cryptology (EUROCRYPT'00)*, pp. 139–155, 2000.

[5] M. Burrows, M. Abadi, R. Needham, ”A logic of authentication,”*ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[6] R. Canetti, H. Krawczyk, ”Analysis of key-exchange protocols and their use for building secure channels,” in *Advances in Cryptology (EUROCRYPT'01)*, pp. 453–474, 2001.

[7] K. Charif, A. Drissi, Z. El A. Guennoun, “A pseudo random number generator based on chaotic billiards,” *International Journal of Network Security*, vol. 19, No. 3, pp. 479-486, 2017.

[8] C. Guo, C. C. Chang, ”Chaotic maps-based password-authenticated key agreement using smart

Table 5: Descriptions the model of Canetti and Krawczyk

Symbol	Definition
parties $P_1, ...P_n$	Modeled by probabilistic Turing machines.
Adversary Λ	A probabilistic Turing machine which controls all communication, with the exception that the adversary cannot inject or modify messages (except for messages from corrupted parties or sessions), and any message may be delivered at most once.
Send query	The adversary can control over Parties outgoing messages via the Send query. Parties can be activated by the adversary launching Send queries.
Corrupt(P_i)	This query allows the adversary to take over the party P_i, including long-lived keys and any session-specific information in P_i memory. A corrupted party produces no further output.
Test(s)	This query allows the adversary to be issued at any stage to a completed, fresh, unexpired session s. A bit b is then picked randomly. If b = 0, the test oracle reveals the session key, and if b = 1, it generates a random value in the key space. The adversary can then continue to issue queries as desired, with the exception that it cannot expose the test session. At any point, the adversary can try to guess b. Let $GoodGuess^{\Lambda}(k)$ be the event that the adversary Λ correctly guesses b, and we define the advantage of adversary Λ as $Advantage^{\Lambda}(k) = \max\{0, \lvert \Pr[GoodGuess^{\Lambda}(k)] - \frac{1}{2} \rvert\}$, where k is a security parameter.

cards,"*Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 6, pp. 1433–1440, 2013.

[9] X. Y. Huang, Y. Xiang, A. Chonka, J. Y. Zhou, R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems,"*IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1390–1397, 2011.

[10] X. Y. Huang, Y. Xiang, E. Bertino, J. Y. Zhou, L.Xu, "Robust multi-factor authentication for fragile communications,"*IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2014.

[11] N. Koblitz, A. Menezes, S. Vanstone, "The state of elliptic curve cryptography,"*Designs, Codes and Cryptography*, vol. 19, no. 2–3, pp. 173–193, 2000.

[12] L. Kocarev, S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*, Springer, 2011.

[13] C. C. Lee, C. T. Li, C. W. Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps,"*Nonlinear Dynamics*, vol. 73, no. 1–2, pp. 125–132. 2013.

[14] H. Lai, M. A. Orgun, J. H. Xiao, J. Pieprzyk, L. Y. Xue, Y. X. Yang, "Provably secure three-party key agreement protocol using Chebyshev chaotic maps in the standard model,"*Nonlinear Dynamics*, vol. 77, no. 4, pp. 1427–1439, 2014.

[15] H. Y. Lin, "Chaotic map based mobile dynamic ID authenticated key agreement scheme,"*Wireless Personal Communications*, vol. 78, no. 2, pp. 1487–1494, 2014.

[16] T. H. Liu, Q. Wang, H. F. Zhu, "A multi-function password mutual authentication key agreement scheme with privacy preserving,"*Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 2, pp. 165–178, 2014.

[17] C. C. Lee, C. T. Li, C. W. Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Nonlinear Dynamics*, vol. 73, no. 1–2, pp. 125–132, 2013.

[18] P. R. Newswire, *Ticketmaster Launches New, Innovative CAPTCHA Solutions, Making The Fan Experience Better*, Jan, 2013. (`http://www.prnewswire.com/news-releases/ticketmaster-launches-new-innovative-captcha-solutions-making-the-fan-experience-better-189000181.htm`)

[19] R. S. Pippal, C. D. Jaidhar, S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[20] A. Stolbunov, "Reductionist security arguments for public-key cryptographic schemes based on group action," in *The Norwegian Information Security Conference (NISK'09)*, pp. 97–109, 2009.

[21] Y. Sun, H. Zhu, and X. Feng, “A novel and concise multi-receiver protocol based on chaotic maps with privacy protection,” *International Journal of Network Security*, vol. 19, No. 3, pp. 371-382, 2017.

[22] B. Wang, M. Ma, "A smart card based efficient and secured multi-server authentication scheme," *Wireless Personal Communications*, vol. 68, no. 2, pp. 361–378, 2013.

[23] G. Wang, J. Yu, Q. Xie, "Security analysis of a single sign on mechanism for distributed computer networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 294–302, 2013.

[24] X. Wang, J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlin-*

ear Science Numerical Simulation, vol. 15, no. 12, pp. 4052–4057, 2010.

[25] H. Wang, H. Zhang, J. Li and C. Xu, “A(3,3) visual cryptography scheme for authentication”, *Journal of Shenyang Normal University (Natural Science Edition)*, vol. 31, no. 101, pp. 397–400, 2013.

[26] Q. Xie, J. M. Zhao, X. Y. Yu, ”Chaotic maps-based three-party password-authenticated key agreement scheme,” *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1021–1027, 2013.

[27] E. J. Yoon, K. Y. Yoo, ”Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem,” *The Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2013.

[28] L. Zhang, ”Cryptanalysis of the public key encryption based on multiple chaotic systems,” *Chaos Solitons Fractals*, vol. 37, no. 3, pp. 669–674, 2008.

[29] H. Zhu, Y. Zhang, and Y. Zhang, “ A provably password authenticated key exchange scheme based on chaotic maps in different realm,” *International Journal of Network Security*, vol. 18, No. 4, pp. 688-698, 2016.

[30] H. Zhu, “Secure chaotic maps-based group key agreement scheme with Privacy Preserving,” *International Journal of Network Security*, vol. 18, No. 6, pp. 1001-1009, 2016.

[31] H. F. Zhu, ”Sustained and authenticated of a universal construction for multiple key agreement based on chaotic maps with privacy preserving,” *Journal of Internet Technology*, vol. 17, no. 5, pp. 1–10, 2015.

[32] H. F. Zhu, M. Jiang, X. Hao, Y. Zhang, ”Robust biometrics-based key agreement scheme with smart cards towards a new architecture,” *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 81–98, 2015.

[33] H. Zhu, Y. Zhang, “An improved two-party password-authenticated key agreement protocol with privacy protection based on chaotic maps,” *International Journal of Network Security*, vol. 19, No. 4, pp. 487-497, 2017.

[34] H. Zhu, Y. Zhang, Y. Xia, and H. Li, “Password-authenticated key exchange scheme using chaotic maps towards a new architecture in standard model,” *International Journal of Network Security*, vol. 18, No. 2, pp. 326-334, 2016.

[35] H. Zhu, Y. Zhang, Y. Zhang and H. Li, “A novel and provable authenticated key agreement protocol with privacy protection based on chaotic maps towards mobile network,” *International Journal of Network Security*, vol. 18, No. 1, pp. 116-123, 2016.

[36] H. Zhu, Y. Zhang, and Y. Sun, “Provably secure multi-server privacy-protection system based,” *International Journal of Network Security*, vol. 18, No. 5, pp. 803-815, 2016.

A Appendix: A case proof for Symmetric Encryption

There are two points of this section should be explained firstly:

1) In order to simplify, we just give the provable security of the first method–Algorithm 1 Symmetric Encryption. And we only consider the one-way secret delivery with privacy preserving between node1 and $node_2$.

2) We only use parts of the adversarial model of Canetti and Krawczyk [6]. The basic descriptions are shown in Table 5.

Definition 4. *A secret messages transfered with privacy preserving protocol* Π_1 *in security parameter* k *is said to be secure in the adversarial model of Canetti and Krawczyk if for any polynomial-time adversary* Λ*,*

Algorithm 1 Symmetric Encryption method simulator

```
Input: $H, E_K()/D_K(), (x, T_{K_2}(x))$
Output: $d$
$r \xleftarrow{R} \{1, ..., k\}$ where $k$ is an upper bound on the number of sessions activated by $\Lambda$ in any interaction.
Invoke $\Lambda$ and simulate the protocol to $\Lambda$, except for the $r-th$ activated protocol session.
for the $r-th$ session, let $node_1$ send $\{i, T_a(x), C\}$ to $node_2$, where $i$ is the session identifier. The $node_2$ can get the secret messages after he authenticate the timestamp and the value of hash in the ciphertext using his own secret key by one-round. do
    if the $r-th$ session is chosen by $\Lambda$ as the test session then
        Provide $\Lambda$ as the answer to the test query.
        $d \xleftarrow{R} \Lambda's$ output.
    else
        $d \leftarrow \{0, 1\}$.
    end if
end for
```

1) If two uncorrupted parties have completed transferring secret messages with privacy preserving by pre-distributed parameter, these sessions produce the same messages as node1s input;

2) $Advantage^{\Lambda}(k)$ is negligible.

Theorem 1. *Under the CDH assumption, using the Algorithm 1 to transfer messages is message-secure in the adversarial model of Canetti and Krawczyk [22].*

Proof. The proof is based on the proof given by Refs. [19, 22]. There are two uncorrupted parties in matching sessions output the same session key, and thus the first part of Definition 4. is satisfied. To show that the second part of the definition is satisfied, assume that there is a polynomial-time adversary Λ with a non-negligible advantage ε in standard model. We claim that Algorithm 1 forms a polynomial-time distinguisher for CDH having non-negligible advantage. □

Probability analysis. It is clear that Algorithm 1 runs in polynomial time and has non-negligible advantage. There are two cases where the r-th session is chosen by Λ as the test session:

1) If the r-th session is not the test session, then Algorithm 1 outputs a random bit, and thus its advantage in solving the CDH is 0.

2) If the r-th session is the test session, then Λ will succeed with advantage ε, since the simulated protocol provided to Λ is indistinguishable from the real protocol. The latter case occurs with probability $1/k$, so the overall advantage of the CDH distinguisher is ε/k, which is non-negligible.

Biography

Hongfeng Zhu obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. He is a full professor of the Kexin software college at Shenyang Normal University. He is also a master's supervisor. He has research interests in wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Dr. Zhu had published more than 50 international journal papers (SCI or EI journals) on the above research fields.

Rui Wang graduated with a Bachelor of Engineering from Shenyang Normal University in 2016. In her college, after completing the learning task, She interests in exploring her professional knowledge. During graduate, under the guidance of his master instructor, she researches information security theory and technology.

A Secure and Efficient Mutual Authentication and Key Agreement Protocol with Smart Cards for Wireless Communications

Cheng Guo[1,2], Chin-Chen Chang[3,4], and Shih-Chang Chang[5]

(Corresponding author: Chin-Chen Chang)

School of Software Technology, Dalian University of Technology, China[1]

Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China[2]

Department of Information Engineering and Computer Science, Feng Chia University, Taiwan[3]

Department of Computer Science and Information Engineering, Asia University, Taiwan[4]

(E-mail: alan3c@gmail.com)

Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan[5]

(Received Oct. 10, 2016; revised and accepted Feb. 20, 2017)

Abstract

Mobile user authentication and key agreement for wireless networks is an important security priority. In recent years, several user authentication and key agreement protocols with smart cards for wireless communications have been proposed. In 2011, Xu et al. proposed an efficient mutual authentication and key agreement protocol with an anonymity property. Although the protocol of Xu *et al.* has many benefits, we find that it still suffers from several weaknesses which have been previously overlooked. In this paper, we propose a secure and efficient mutual authentication and key agreement protocol. Confidentiality of the session key and updating of the password efficiently are presented as the main contributions of this paper. Finally, evaluations of our proposed protocol show that our protocol can withstand various known types of attacks, and also satisfies essential functionality requirements. Additionally, efficiency analyses show that our protocol is simple and cost-efficient.

Keywords: Authentication; Key Agreement; Smart Cards; Wireless Communications

1 Introduction

With the rapid development of wireless communications technologies, devices, such as the mobile phone, the PDA (personal digital assistant), and the iPad, have become more and more popular. Mobile users can roam into a foreign network and transmit messages or data to foreign agents, or access some services provided by a foreign agent by using their mobile devices. Obliviously, before establishing communication between a mobile user and a foreign agent, the foreign agent and the user need to carry out mutual authentication and establish a session key. Confidentiality and authentication are two fundamental security goals for wireless communications. Up to now, many authentication and key agreement schemes for wireless environments have been proposed in the literature [3, 7, 8, 16, 19, 23]. Due to the limited power consumption, bandwidth, and storage resources of mobile devices, the authentication and key agreement protocol must take computation efficiency and communication cost on mobile users into consideration. Currently, smart card based user authentication schemes [2, 4, 6, 10, 13, 14, 17, 20, 21] have been widely developed and applied due to their low computational cost, convenient portability, and cryptographic properties. Therefore, in recent years, research [1, 3, 9, 10, 18, 19, 22] on authentication and key agreement based on smart cards for wireless communications has become more and more popular in the world.

In 1998, Miller *et al.* [12] first proposed an authentication and key agreement protocol based on elliptic curve cryptograph (ECC). Their scheme is suitable for running in wireless mobile devices due to its low computational cost. In 2004, Zhu and Ma [23] proposed an authentication protocol with anonymity for wireless communications using smart cards. Later, Lee *et al.* [7] pointed out that Zhu and Ma's protocol has several security weaknesses, and then they improved it. However, recently, Xu *et al.* [19] showed that Lee *et al.*'s protocol [7] is vulnerable to several weaknesses, and then proposed a mutual authentication and key agreement protocol preserving user anonymity in mobile networks. They claimed that their protocol was immune to various known types of attacks and achieved identity anonymity, key agreement fairness, and user friendliness. However, in this paper, we will show that Xu *et al.*'s protocol has three weaknesses as follows:

1) It cannot protect against an insider attack;

2) Session-key problem;

3) Inefficiency of the password-changing operation;

We will detail these weaknesses later.

To the best of our knowledge, in the most existing authentication and key agreement protocols for wireless communications, the home agent was able to compute the session key between the mobile user and the foreign agent. There are potential risks for some confidential communications.

In order to remedy these weaknesses, we propose an enhanced authentication and key agreement protocol with smart cards for wireless communications. Analysis shows that the proposed protocol is effective in protection from the above weaknesses. Furthermore, our protocol is immune to various known types of attacks. Compared with the previous schemes [3, 7, 19], our protocol not only satisfies more security and functionality requirements, but also provides an acceptable computational cost. As mentioned in the literature [3, 5, 9, 15, 18, 19] and the above description, the following requirements are important for a strong user mutual authentication and key agreement protocols using passwords and smart cards for wireless communications:

1) Mutual authentication;

2) Update password freely and efficiently;

3) Fairness in key agreement;

4) No registration/password table;

5) Low communication cost and computational complexity;

6) Protection of user anonymity;

7) Withstanding the insider attack;

8) Withstanding the replay attack;

9) Withstanding the offline dictionary attack without the smart card;

10) Withstanding the offline dictionary attack with the smart card;

11) Confidentiality of the session key.

The reminder of this paper is organized as follows. In next section, we briefly review Xu *et al.*'s scheme, whose weaknesses are pinpointed in Section 3. Section 4 describes a new secure and efficient mutual authentication and key agreement protocol that gives a remedy for Xu *et al.*'s scheme. Section 5 analyzes the security of our protocol and presents functionality considerations and computational cost among our protocol and the related protocols. Finally, we give a conclusion in Section 6.

2 Review of Xu *et al.*'s Scheme

In this section, we review Xu *et al.*'s [19] mutual authentication and key agreement protocol in mobile networks, which claims to be immune to various known types of attacks. Basically, Xu *et al.*'s scheme contains three phases: out-of-band registration, mutual authentication between MN and FA, and session key renewal. Table 1 lists the notations used in Xu *et al.*'s scheme.

Table 1: The notations used in Xu *et al.*'s scheme

Notations	Descriptions
MN	A mobile user
HA	Home agent of a mobile user
FA	Foreign agent of the network visited by the user
PW_X	The password of a mobile user X
ID_X	The identity of an entity X
$h(.)$	A one-way hash function
T_X	Time stamp by an entity X
$\|\|$	String concatenation operation
$\oplus$	The bitwise XOR operation
$E_{K(.)}$	Symmetric encryption of a message using key K

2.1 Out-of-band Registration Phase

Xu *et al.*'s scheme is initialized by the HA with choosing the public parameters (p, q, g), where g is a generator in a multiplicative group of order q, $p = 2q+1$ is the modulus for the group, and both p and q are public large prime numbers. The HA selects a private key b and computes its public key $B = g^b \bmod p$. When a mobile user MN wants to register with his HA, he chooses his identity ID_{MN} and password PW_{MN}, and then submits them to the HA. Then HA computes $u = E_N(h(PW_{MN})||ID_{MN})$ with its server secret key N, and securely issues a smart card to the MN, which contains p, g, B, u, and $h(.)$.

2.2 Mutual Authentication Phase

As a precondition, HA needs to pre-share a distinct symmetric key k_{FH} with each FA. When a user MN visits a new foreign network, the following steps are performed:

Step 1. MN enters his identity ID_{MN} and his password PW_{MN} to the smart card. Then the device selects two secret random numbers a and r_{MN}, computes $A = g^a \bmod p$, $D = h(B^a \bmod p)$, $C = h(PW_{MN}) \oplus D$, $R = E_C(r_{MN}||ID_{MN})$, and $U = E_D(T_{MN}||u)$, where T_{MN} is a time stamp. Finally, the device sends the authentication message $(ID_{HA}, T_{MN}, A, R, U)$ to FA.

Step 2. On receiving the authentication message, FA first checks whether T_{MN} is valid. If so, FA chooses a random number r_{FA}, generates its timestamp T_{FA}, computes $Q = E_{k_{FH}}(T_{MN}||r_{FA}||T_{FA}||U||R)$, and sends the message (A, Q) to HA.

Step 3. On receiving the message (A, Q), HA decrypts Q with k_{FH}, and obtains the message $T_{MN}||r_{FA}||T_{FA}||U||R$. HA first checks whether T_{FA} is valid. If so, HA computes $D = h(A^b \bmod p)$ for decrypting U, so as to recover $T_{MN}||u$. Then HA checks whether the T_{MN} equals the previous one from Q. If the timestamp is valid, HA decrypts u with his secret key N to recover $h(PW_{MN})||ID_{MN}$, and computes $C = h(PW_{MN}) \oplus D$ so as to recover $r_{MN}||ID_{MN}$. If ID_{MN} from R equals the previous one from u, MN is authenticated. Finally, HA computes $K = E_{k_{FH}}(r_{FA}||r_{MN})$ and $S = E_D(r_{MN}||r_{FA}||ID_{FA})$, and sends the message (K, S) to FA.

Step 4. FA decrypts K to obtain $r_{FA}||r_{MN}$. If the recovered r_{FA} equals the previous one, the FA believes that MN is an authorized user, and forwards S to MN.

Step 5. MN decrypts S to recover $r_{MN}||r_{FA}||ID_{FA}$. If r_{MN} and ID_{FA} are verified, MN believes FA is authenticated. Then, both MN and FA can compute the session key $k = r_{MN} \oplus r_{FA}$.

2.3 Session Key Renewal Phase

In this phase, the session key can be updated by MN and FA from k_i for the current ith session to $k_{i+1} = h(k_i||r_{MN})$ for the next session.

3 Weaknesses of Xu *et al.*'s Scheme

The authors of [19] proposed a mutual authentication and key agreement protocol featuring user identity anonymity. They claimed that their protocol was resilient to various attacks, cost-efficient for a mobile user, and achieved key agreement fairness. However, in this section, we present that Xu *et al.*'s scheme still has a number of serious deficiencies. The detailed description of three weaknesses is as follows.

3.1 Insider Attack

Insider attack means that if an insider of HA has obtained a mobile user MN's password, there may be an unauthorized and illegitimate access to any foreign agent. In the registration phase of Xu *et al.*'s protocol, MN sent his identity ID_{MN} and password PW_{MN} to the home agent HA, that is, PW_{MN} was revealed to HA. It will give an inside attacker an opportunity to impersonate other users. With a strong authentication protocol in wireless networks, there should be no way to directly obtain the users' passwords.

3.2 Session-key Problem

In Xu *et al.*'s protocol, the session key can be computed as $k = r_{MN} \oplus r_{FA}$ by MN and FA, respectively. However, the mutual authentication between MN and FA must resort to the assistance of the corresponding HA. However, in the mutual authentication phase of Xu *et al.*'s protocol, HA can recover the r_{MN} and r_{FA} from the authentication request. Therefore, HA also has a capability to compute the session key $k = r_{MN} \oplus r_{FA}$. It is well known that this session key is used to encrypt all messages in the communication session between MN and FA. We can see that the protocol of Xu *et al.* merely provided the session key agreement. However, they cannot guarantee the confidentiality of the session key. As to many applications, there are potential risks. That is, the design in Xu *et al.*'s protocol is actually insecure and infeasible.

3.3 Inefficiency of Password-changing Operation

In a strong user authentication and key agreement protocol, user should have a capability to update his password. In Xu *et al.*'s protocol, we can see that HA computes $u = E_N(h(PW_{MN})||ID_{MN})$ with its server secret key N, and stores u into the smart card. It means that the smart card needs to replace the old parameter u with the new parameter $u*$. Since the password PW_{MN} is encrypted by HA using his secret key N, MN has to resubmit his new PW^*_{MN} to HA, and HA replaces the original u in MN's smart card with $u* = E_N(h(PW^*_{MN})||ID_{MN})$. MN has to communicate with HA through the smart card and the wireless network. This makes the password-changing operation inefficient.

4 The Proposed Protocol

To overcome the above-mentioned weaknesses, in this section, we propose a secure and efficient mutual authentication and key agreement protocol with smart cards for wireless communications, which consists of parameter-generation phase, registration phase, authentication phase, key agreement phase, and password-change phase.

4.1 Parameter-generation Phase

Compared with public key cryptosystems such as RSA and ElGamal, ECC provides a better efficiency because it is believed that the same level of security can be achieved with a smaller key size. Therefore, ECC-based authentication and key agreement protocols are more suitable for smart cards and mobile devices which usually have energy constraints and significant bandwidth.

For ease of presentation, we employ some abbreviations and notations provided in Xu *et al.*'s protocol, summarized in Table 1. In the proposed protocol, the elliptic curve equation is defined as $E_P : y^2 = x^3 + ax + b(\bmod p)$

over a prime finite field Z_p, where $a, b \in Z_p$, and $4a^3 + 27b^2(\mathrm{mod}p) \neq 0$.

Our protocol is initialized by the HA by choosing the public parameters (E_P, G), where G is a generator point of E_P. The HA selects his private key $SK_{HA} = c$ and computes his public key $PK_{HA} = c \times G(\mathrm{mod}p)$. Also, the FA selects his private key $SK_{FA} = d$ and computes his public key $PK_{FA} = d \times G(\mathrm{mod}p)$.

4.2 Registration Phase

The registration phase is similar to that of Xu *et al.*'s protocol. When a mobile user MN wants to register and become a new legal user, MN submits his identity ID_{MN} to HA over a secure channel. As is shown in Figure 1, the following steps are performed by the HA in this phase.

Step 1. HA computes $u = E_{K_S}(ID_{MN}||K_S) \oplus h(pw)$, where K_S is the HA's secret key, and pw is the initial password selected by HA.

Step 2. HA computes $IM = E_{K_S}(ID_{MN}||r)$, where r is a random number to provide the identity protection.

Step 3. HA issues the password pw and the smart card to MN over a secure channel, where the smart card contains $\{u, IM, h(.), PK_{HA}\}$.

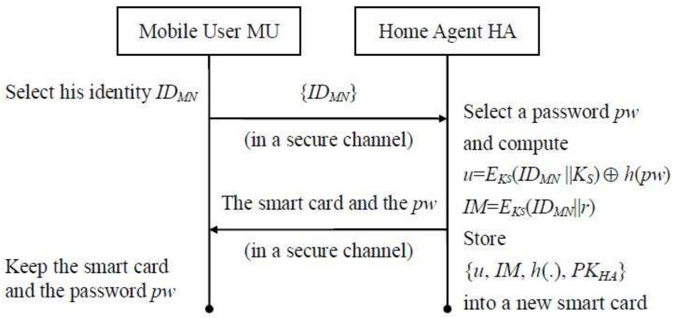


Figure 1: The registration phase of the proposed protocol

4.3 Authentication Phase

In this phase, the mobile user MN and a foreign agent FA can authenticate each other and share a session key K for the subsequent secret communication. When a mobile user MN roams into a new foreign network and wants to access service, the following steps are performed.

Step 1. MN enters his identity ID_{MN} and his password pw to the smart card. Then the smart card chooses two random numbers e and r_{MN}, and computes $A = e \times G(\mathrm{mod}p)$ and $U = E_D(T_{MN}||u)$, where T_{MN} is a current time stamp and $D = e \times PK_{HA} = e \times c \times G$. Subsequently, the smart card computes $N = h(pw) \oplus D$ as a secret key, and computes $R = E_N(r_{MN}||ID_{MN})$.

Step 2. The smart card sends an authentication request message $\{R, ID_{HA}, IM, U, A, T_{MN}\}$ to FA.

Step 3. After receiving the authentication request from MN, FA first check whether the time stamp T_{MN} is valid. If it is valid, FA selects a random number r_{FA} and computes $V = E_M(T_{MN}||r_{FA}||T_{FA}||U||IM||R)$, where $M = d \times PK_{HA} = d \times c \times G$, and T_{FA} is a time stamp. Then, FA sends the message $\{A, V, PK_{FA}\}$ to HA.

Step 4. After receiving the message $\{A, V, PK_{FA}\}$, HA first computes $M = c \times PK_{FA} = c \times d \times G$, and decrypts V by using M. HA then verifies whether T_{FA} is valid. If so, HA decrypts IM by using his secret key K_S to recover ID_{MN}, and computes $E_{K_S}(ID_{MN}||K_S)$. Then, HA computes $D = c \times A = c \times e \times G$ for decrypting U aiming at recovering T_{MN} and u. HA checks whether the decrypted T_{MN} is the same as the T_{MN} decrypted from V. If they are valid, HA can obtain $h'(pw)$ by computing $E_{K_S}(ID_{MN}||K_S) \oplus u$.

Step 5. HA can compute $N' = h'(pw) \oplus D$. If MN submitted a correct password in Step 1, we can believe $N = N'$. And, HA can decrypt R for recovering r_{MN} and ID_{MN}. Then HA checks whether ID_{MN} from R equals the previous one from u. If so, MN is authenticated.

4.4 Key Agreement Phase

In this phase, FA and HA can be authenticated, and a secure session key between MN and FA can be established. The steps of this phase are shown as follows.

Step 1. HA computes $K = E_{cPK_{FA}}(r_{MN}||r_{FA})$ and $S = E_{cA}(r_{MN}||r_{FA}||ID_{FA}||PK_{FA})$, and sends the message (K, S) to FA.

Step 2. After receiving the message, FA decrypts K with $d \times PK_{HA}$ to recover r_{MN} and r_{FA}. If the recovered r_{FA} and the original one are identical, FA believes that the mobile user MN is a valid user. Then FA forwards S to MN.

Step 3. Upon receiving the message S from FA, MN first decrypts S with $e \times PK_{HA}$, where $e \times PK_{HA} = c \times A$, for recovering $\{r_{MN}, r_{FA}, ID_{FA}, PK_{FA}\}$. If the r_{MN} and the ID_{FA} are both verified, MN believes that FA is authenticated. Finally, both MN and FA can compute the agreed session key $SK = E_{d \times A}(r_{MN} \oplus r_{FA}) = E_{e \times PK_{FA}}(r_{MN} \oplus r_{FA})$.

The above two phases are outlined in Figure 2.

4.5 Password-change Phase

When a mobile user MN wants to renew a password, MN can insert his smart card into the card reader and performs the following steps.

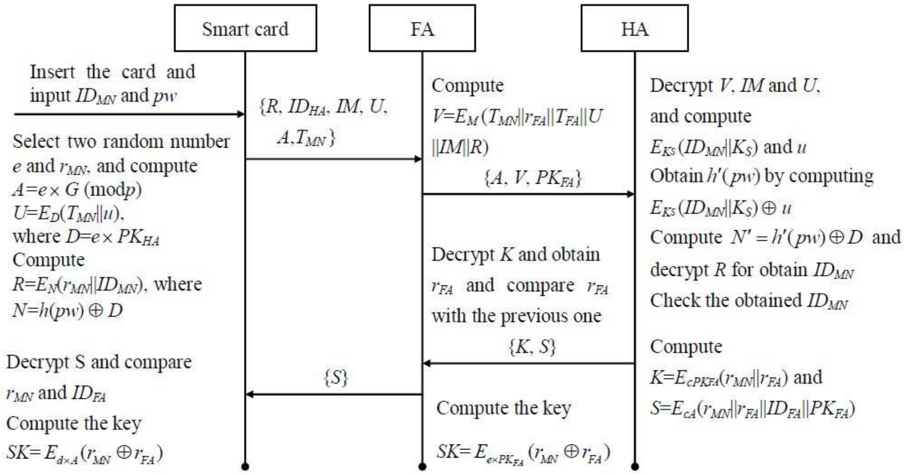

Smart card
FA
HA
Insert the card and input ID_{MN} and pw
Select two random number e and r_{MN}, and compute $A=e\times G$ (mod$p$) $U=E_D(T_{MN}||u)$, where $D=e\times PK_{HA}$
Compute $R=E_N(r_{MN}||ID_{MN})$, where $N=h(pw)\oplus D$
$\{R, ID_{HA}, IM, U, A, T_{MN}\}$
Compute $V=E_M(T_{MN}||r_{FA}||T_{FA}||U||IM||R)$
$\{A, V, PK_{FA}\}$
Decrypt V, IM and U, and compute $E_{KS}(ID_{MN}||K_S)$ and u
Obtain $h'(pw)$ by computing $E_{KS}(ID_{MN}||K_S)\oplus u$
Compute $N'=h'(pw)\oplus D$ and decrypt R for obtain ID_{MN}
Check the obtained ID_{MN}
Compute $K=E_{cPK_{FA}}(r_{MN}||r_{FA})$ and $S=E_{cA}(r_{MN}||r_{FA}||ID_{FA}||PK_{FA})$
$\{K, S\}$
Decrypt K and obtain r_{FA} and compare r_{FA} with the previous one
Compute the key $SK=E_{e\times PK_{FA}}(r_{MN}\oplus r_{FA})$
$\{S\}$
Decrypt S and compare r_{MN} and ID_{FA}
Compute the key $SK=E_{d\times A}(r_{MN}\oplus r_{FA})$

Figure 2: The authentication and key agreement phase of the proposed protocol

Step 1. Firstly, MN keys in the old password, and requests to renew password pw. Next, MN enters the new password pw^*.

Step 2. The smart card computes $u* = u \oplus h(pw) \oplus h(pw*)$, and replaces u with $u*$.

5 Evaluations of the Proposed Protocol

In this section, we will give an analysis for our protocol in terms of security, functionality, and efficiency.

5.1 Security Analysis

In this subsection, security analysis of our protocol will be discussed. At the end of this subsection, the comparisons of the related works are given in Table 2.

1) User anonymity

Anonymity is becoming a major concern in many security requirements. The aim of user anonymity in wireless networks is to make sure that the real identity of a mobile user is protected from anyone besides his home agent.

During the registration phase, HA computes u by encrypting $ID_{MN}||K_S$ with his secret key K_S and computes $IM = E_{K_S}(ID_{MN}||r)$, where r is a random number to provide the identity protection. Even though the smart card containing u and IM is lost and the attackers obtain $\{u, IM\}$ from the smart card, they can not retrieve any information about the user's real identity since the identity information is encrypted by using HA's secret key K_S. In the authentication phase, the smart card sends an authentication request $\{R, ID_{HA}, IM, U, A, T_{MN}\}$ to FA. Since R and U are encrypted by using the corresponding session key D between MN and HA and N, where $N = h(pw) \oplus D$, respectively, HA cannot retrieve the real identity of the user according to these authentication messages.

2) Withstanding the insider attack

The insider attack in the authentication for wireless networks is that the mobile user's password is submitted to the home agent in the registration phase, and an inside attacker can steal his password and impersonate this user. In our proposed protocol, in the registration phase, the initial password is provided by the home agent. And the home agent issues a smart card containing this password and other security parameters to the mobile user. And then, the user has a capability to change the password freely and securely. As described in Section 4.5, the update-password operation does not require the assistance of the home agent. Therefore, an inside attacker cannot obtain any information about the user's password from the home agent.

3) Withstanding the replay attack

The replay attack is that the login messages are maliciously or fraudulently repeated or delayed between the user and the server. In the authentication phase, the authentication request message contains the timestamp T_{MN} and T_{FA} . If the adversaries delay the request message or resend these messages, we can check whether the current time is valid according to the timestamp.

4) Withstanding the offline dictionary attack without the smart card

The offline dictionary attack without the smart card is that attackers attempt to guess the user's password or the user's identity via the intercepted message between the mobile user and the foreign agent or between the foreign agent and the home agent. In the proposed protocol, we can see that the authentication request message $R = E_N(r_{MN}||ID_{MN})$,

Table 2: Security comparisons between the related protocols and the proposed protocol

	Our protocol	**He *et al.*'s protocol [3]**	**Xu et la.'s protocol [19]**	**Lee *et al.*'s protocol [7]**
S1	YES	NO	YES	NO
S2	YES	YES	NO	NO
S3	YES	YES	YES	YES
S4	YES	YES	YES	YES
S5	YES	YES	YES	YES
S6	YES	NO	NO	NO

S1: User anonymity; S2: Withstanding the insider attack; S3: Withstanding the replay attack; S4: Withstanding the offline dictionary attack without the smart card; S5: Withstanding the offline dictionary attack with the smart card; S6: Confidentiality of the session key

$IM = E_{K_S}(ID_{MN}||r)$ and $U = E_D(T_{MN}||u)$. Since r_{MN} and r are random numbers, and $u = E_{K_S}(ID_{MN}||K_S) \oplus h(pw)$, even though the identity ID_{MN} and the password are weak, the attackers cannot guess the right password and the identity. In the next phase, FA sends the message $\{A, V, PK_{FA}\}$ to HA, where V contains a random number r_{FA}. So, the attackers cannot obtain any information about the user's identity and his password.

5) Withstanding the offline dictionary attack with the smart card

We assume that the attackers can obtain the information stored in the smart card through some ways. This attack is the same as the above attack except in this case. In our proposed protocol, the password pw is stored in the smart card as the form $u = E_{K_S}(ID_{MN}||K_S) \oplus h(pw)$. Since the attackers cannot obtain the home agent's secret key K_S, they cannot guess the right password.

6) Confidentiality of the session key

Key agreement protocol is to establish a secret session key between the mobile user MN and the foreign agent FA aim at encrypting further communications between the MN and the FA. Therefore, the session key should be shared only between the MN and the FA, and other entities cannot retrieve any information about the session key. However, in [3, 7, 19], the home agent also has a capability to calculate the session key. In the proposed protocol, the session key $SK = E_{d\times A}(r_{MN} \oplus r_{FA}) = E_{e\times PK_{FA}}(r_{MN} \oplus r_{FA})$, where the $r_{MN} \oplus r_{FA}$ is encrypted by using an agreed key. Though the home agent can obtain r_{MN} and r_{FA}, he can not compute the key $d\times e\times G = e\times d\times G$. So, HA cannot compute the session key SK.

5.2 Functionality Consideration

Now we move on the functionality consideration. Some comparisons with related works are presented. We examine our protocol as follows.

1) Mutual authentication

In the proposed protocol, the foreign agent and the mobile user can authenticate each other under the assistance of the home agent HA, and after a successful validation, a common session key can be established between the foreign agent and the mobile user.

To achieve the mutual authentication, HA stores $u = E_{K_S}(ID_{MN}||K_S) \oplus h(pw)$ and $IM = E_{K_S}(ID_{MN}||r)$ in MN's smart card at the registration phase. In Step 1 of the authentication phase, the smart card computes $R = E_N(r_{MN}||ID_{MN})$, where $N = h(pw) \oplus D$, and sends $\{R, IM, U\}$ to FA. FA utilizes a session key M between FA and HA to encrypt these authenticated request information $\{R, IM, U\}$ and sends these messages to HA. HA can recover $\{R, IM, U\}$ and retrieve ID_{MN} from IM by using his secret key K_S. Further, HA computes $E_{K_S}(ID_{MN}||K_S)$ and obtain $h'(pw)$ by computing $E_{K_S}(ID_{MN}||K_S) \oplus u$. Then, HA can compute $N' = h'(pw) \oplus D$, and retrieve ID_{MN} from R. We can see that if the mobile user does not submit the correct ID_{MN} and password, HA cannot decrypt R correctly, and the retrieved ID_{MN} cannot equal the previous one from IM. Finally, both MN and FA can compute the agreed session key $SK = E_{d\times A}(r_{MN} \oplus r_{FA}) = E_{e\times PK_{FA}}(r_{MN} \oplus r_{FA})$.

2) Update password freely and efficiently

In the proposed protocol, when the home agent HA issues a smart card to the mobile user MN, the MN can update his password freely. As described in Section 4.5, each mobile user can choose or update one of his favorite strings as his password, not decided by the home agent. And, the update-password operation can be performed without the assistance of the home agent. As to wireless networks, when a mobile user roams into a foreign network, the update-password operation of our proposed protocol is more suitable than Xu *et al.*'s protocol in terms of communication cost.

3) Fairness in key agreement

Table 3: Functionality comparisons between the related protocols and the proposed protocol

	Our protocol	**He *et al.*'s protocol [3]**	**Xu et la.'s protocol [19]**	**Lee *et al.*'sprotocol [7]**
F1	YES	YES	YES	YES
F2	YES	YES	YES	NO
F3	YES	YES	NO	NO
F4	YES	NO	YES	YES
F5	YES	YES	YES	YES
F1: Mutual authentication; F2: Update password freely; F3: Update password efficiently; F4: Fairness in key agreement; F5: No registration/password table				

Our proposed protocol ends up with MN and FA agreeing on a session key $SK = E_{d\times e\times G}(r_{MN} \oplus r_{FA})$, where r_{MN} and r_{FA} are random numbers selected by MN and FA, respectively, which has a similar structure with the session key of Xu *et al.*'s protocol. The difference is that $r_{MN} \oplus r_{FA}$ is encrypted by using an agree key $d \times A = e \times PK_{FA}$, where e is a secret value of MN, and d is a secret value of FA. Therefore, the session key SK contains equal contributions from both parties.

4) No registration/password table

In some password-based authentication protocols, the server has to store a password table or a registration table for verification. That is, HA has to maintain a secret and large table. And, it will give a chance for an inside attacker to access the password or the registration information. In the proposed protocol, HA does not need to keep a registration table or a password table. HA can compute the verification messages $u = E_{K_S}(ID_{MN}||K_S) \oplus h(pw)$ and $IM = E_{K_S}(ID_{MN}||r)$, and store u and IM into the smart card. In the authentication phase, FA and MN can authenticate each other under the assistance of HA.

Finally, we summarize the functionality of our protocol and make comparisons with that of related works [3, 7, 19] in Table 3.

5.3 Efficiency Analysis

In this subsection, we will evaluate the performance of our protocol and make comparisons with the related works. The heavy weight computation of the proposed protocol is to execute scalar multiplication on elliptic curve. The computational cost of elliptic curve point multiplication is much less than that of modular exponentiation, and 160-bit elliptic curve discrete logarithm problem and 1024-bit discrete logarithm problem have the same security level. Therefore, contrary to traditional public key cryptosystem-based authentication and key agreement protocol, our proposed protocol reduces the computation, communication, and storage space costs since ECC is used. In Table 4, we tabulate the computational costs of MN, FA, and HA for our proposed protocol and the related protocols [3, 7, 19].

6 Conclusions

Recently, Xu *et al.* proposed a mutual authentication and key agreement protocol preserving user anonymity in mobile networks. In this paper, we have shown the weaknesses of the protocol of Xu *et al.* and further proposed an improved protocol based on ECC. Furthermore, we propose a secure and efficient mutual authentication and key agreement protocol with smart cards for wireless communications. Security analyses show that our protocol is able to provide mutual authentication and key agreement with user anonymity and is effective in withstanding various attacks. Meanwhile, our protocol provides essential functionalities that satisfy the most important applications for mobile devices in wireless networks. Efficiency analyses also demonstrate that our protocol is more efficient than the previous ones.

Acknowledgments

This paper is supported by the National Science Foundation of China under grant No. 61401060, 61501080 and 61572095, and the Fundamental Research Funds for the Central Universities' under grant No. DUT16QY09.

References

[1] P. Y. Cui, "An improved ownership transfer and mutual authentication for lightweight RFID protocols," *International Journal of Network Security*, vol. 18, no. 6, pp. 1173-1179, 2016.

[2] C. Guo and C. C. Chang, "Chaotic maps-based password-authenticated key agreement using smart cards," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, pp. 1433–1440, 2013.

[3] D. J. He, M. D. Ma, Y. Zhang, C. Chen, and J. J. Bu, "A strong user authentication scheme with anonymity for wireless communications," *Computer Communications*, vol. 34, pp. 367–374, 2011.

Table 4: Performance comparisons between the related protocols and the proposed protocol

Primitives		**Our protocol**	**He *et al.*'s protocol [3]**	**Xu et la.'s protocol [19]**	**Lee *et al.*'s protocol [7]**
H	MN	1	10	1	2
	FA	N/A	5	N/A	N/A
	HA	1	5	N/A	3
E	MN	N/A	N/A	2	N/A
	FA	N/A	N/A	N/A	N/A
	HA	N/A	N/A	1	N/A
S	MN	4	2	3	2
	FA	3	1	2	1
	HA	7	2	6	1
M	MN	3	N/A	N/A	N/A
	FA	2	3	N/A	N/A
	HA	2	3	N/A	N/A

H: Hash operation; E: Modulus exponential operation; S: Symmetric encryption or decryption; M: Scalar multiplication on elliptic curve

[4] M. S. Hwang, S. K. Chong, and T. Y. Chen, "DoS resistant ID-based password authentication scheme using smart cards," *Journal of Systems and Software*, vol. 83, pp. 163–172, 2010.

[5] M. H. Ibrahim, "Octopus: An edge-fog mutual authentication scheme," *International Journal of Network Security*, vol. 18, no. 6, pp. 1089-1101, 2016.

[6] W. S. Juang, S. T. Chen, and H. T. Liaw, "Robust and efficient password-authenticated key agreement using smart card," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2551–2556, 2008.

[7] C. C. Lee, M. S. Hwang, and I. E. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless environments," *IEEE Transactions on Industrial Electronic*, vol. 53, pp. 1683–1686, 2006.

[8] C. T. Li and M. S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks," *Information Sciences*, vol. 181, pp. 5333–5347, 2011.

[9] C. T. Li and C. C. Lee, "A novel user authentication and privacy preserving scheme with smart cards for wireless communications," *Mathematical and Computer Modeling*, vol. 55, pp. 35–44, 2012.

[10] X. Li, J. W. Niu, M. K. Khan, and J. G. Liao, "An enhanced smart card based remote user password authentication scheme," *Journal of Network and Computer Applications*, vol. 36, pp. 1365–1371, 2013.

[11] Y. Ma, "NFC communications-based mutual authentication scheme for the internet of things," *International Journal of Network Security*, vol. 19, no. 4, pp. 631-638, 2017.

[12] V. S. Miller, "Use of elliptic curves in cryptography," in *Proceeding of the Advances in Cryptology (CRYPTO'85)*, pp. 417–426, 1985.

[13] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure user anonymity-preserving biometric- based multi-server authenticated key agreement scheme using smart cards," *Expert Systems with Applications*, vol. 41, pp. 8129–8143, 2014.

[14] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure and efficient ecc-based user anonymity-preserving session initiation authentication protocol using smart card," *Peer-to-Peer Networking and Applications*, vol. 9, pp. 171–192, 2016.

[15] M. Ramadan, F. Li, C. Xu, *et al.*, "User-to-user mutual authentication and key agreement scheme for LTE cellular system," *International Journal of Network Security*, vol. 18, no. 4, pp. 769-781, 2016.

[16] R. V. Sampangi and S. Sampalli, "Metamorphic framework for key management and authentication in resource-constrained wireless networks," *Information Sciences*, vol. 19, pp. 430–442, 2017.

[17] D. Z. Sun, J. P. Huai, J. Z. Sun, J. X. Li, J. W. Zhang, and Z. Y. Feng, "Improvements of juang *et al.*s password-authenticated key agreement scheme using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 2284–2291, 2009.

[18] C. C. Wu, W. B. Lee, and W. J. Tsaur, "A secure authentication scheme with anonymity for wireless communications," *IEEE Communications Letters*, vol. 12, pp. 722–723, 2008.

[19] J. Xu, W. T. Zhu, and D. G. Feng, "An efficient mutual authentication and key agreement protocol preserving user anonymity in mobile networks," *Computer Communications*, vol. 34, pp. 319–325, 2011.

[20] H. M. Yang, Y. X. Zhang, Y. Z. Zhou, and et al., "Provably secure three-party authenticated key agreement protocol using smart cards," *Computer Networks*, vol. 58, pp. 29–38, 2014.

[21] K. H. Yeh, C. Su, N. W. Lo, Y. Li, and Y. X. Hung, "Two robust remote user authentication protocols using smart cards," *Journal of Systems and Software*, vol. 83, pp. 2556–2565, 2010.

[22] E. J. Yoon, K. Y. Yoo, and K. S. Ha, "A user friendly authentication scheme with anonymity for wireless communications," *Computers and Electrical Engineering*, vol. 37, pp. 356–364, 2011.

[23] J. Zhu and J. Ma, "A new authentication scheme with anonymity for wireless environments," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 230–234, 2004.

Biography

Cheng Guo received the B.S. degree in computer science from Xian University of Architecture and Technology in 2002. He received the M.S. degree in 2006 and his Ph.D in computer application and technology, in 2009, both from the Dalian University of Technology, Dalian, China. From July 2010 to July 2012, he was a post doc in the Department of Computer Science at the National Tsing Hua University, Hsinchu, Taiwan. Since 2013, he has been an associate professor in the School of Software Technology at the Dalian University of Technology. His current research interests include information security, cryptology and cloud security.

Shih-Chang Chang received his B.S. degree in 2005 and his M.S. degree in 2007, both in Department of Information Engineering and Computer Science from Feng Chia University, Taichung, Taiwan. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from National Chung Cheng University, Chiayi, Taiwan. His current research interests include electronic commerce, information security, computer cryptography, and mobile communications.

Chin-Chen Chang received his B.S. degree in applied mathematics in 1977 and the M.S. degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. From 1983-1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From August 1989 to July 1992, he was the head of, and a professor in, the Institute of Computer Science and Information Engineering at the National Chung Cheng University, Chiayi, Taiwan. From August 1992 to July 1995, he was the dean of the college of Engineering at the same university. From August 1995 to October 1997, he was the provost at the National Chung Cheng University. From September 1996 to October 1997, Dr. Chang was the Acting President at the National Chung Cheng University. From July 1998 to June 2000, he was the director of Advisory Office of the Ministry of Education of the R.O.C. From 2002 to 2005, he was a Chair Professor of National Chung Cheng University. Since February 2005, he has been a Chair Professor of Feng Chia University. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression and data structures. He is a fellow of the IEEE.

A Generalize Estimating the $\phi(n)$ of Upper/Lower Bound to RSA Public Key Cryptosystem*

Jie Fang[1], Chenglian Liu[2]
(Corresponding author: Chenglian Liu)

School of Electronics and Information Engineering, Fuqing Branch of Fujian Normal University[1]
Department of Computer Science and Technology, Neusoft Institute of Guangdon[2]
Nanhai Software and Technology Park, Foshan 528225, Guangdong, China
(Email: chenglian.liu@gmail.com)

(Received Sept. 18, 2016; revised and accepted Jan. 15, 2017)

Abstract

The RSA-768 (270 decimal digits) was factored by Kleinjung *et al.* on December 12, 2009, while the RSA-704 (212 decimal digits) was factored by Bai *et al.* on July 2, 2012. The RSA-200 (663 bits) was factored by Bahr *et al.* on May 9, 2005, while he RSA-210 (696 bits) was factored by Propper on September 26, 2013. In this paper the author will discuss an estimation method to approach the lower/upper bound of $\phi(n)$ to the RSA parameters. Our contribution may help researchers lock the (n) and the challenge RSA shortly.

Keywords: Euler's Totient Function; Factoring; RSA Cryptosystem

1 Introduction

Challenge RSA [19] is a good work to study. Recently, most scientists and researchers [2, 7, 12] using the general number field sieve (GNFS) algorithm to factor RSA modulus n. In a practical environment, it looks like if you want to break the RSA, the best choice is to choose GNFS if you have already factored the modulus n [5]. In theory, Wiener [24] first proposed a cryptanalysis of short secret exponents where the $d < N^{0.5}$ in 1990. Boneh [3] presented 'Twenty years of attacks on RSA cryptosystem' in 1999, where he classified and described a variety of attacks. Followed by Boneh and Durfee [4], they suggested the private key d should be greater than $N^{0.292}$ for the security problem. Even though, some people like to focus on secret key d or factor composite number n. Their purposes are clear. We believe that there must be a general way to estimate the value of RSA-210 without finding the factors of prime numbers p and q to challenge RSA. According to the latest news, the RSA-210 was factored by Propper [18], and RSA-220 was factored by Bai *et al.* [1]. In this article, the author will introduce a new methodology where we approach the lower bound and the upper bound of $\phi(n)$. For this general concept, it may match any bit length composite number n.

2 Review of RSA Conception

The signer prepares the prerequisite of a RSA signature: Two distinct large prime p and q, $n = pq$, Let e be a public key so that $\gcd(e, \phi(n)) = 1$, where $\phi(n) = (p-1)(q-1)$, then calculate the private key d such that $ed \equiv 1 \pmod{\phi(n)}$. The signer publishes (e, n) and keeps (p, q, d) secretly. The notation as same in [19].

2.1 RSA Encryption and Decryption

In RSA public-key encryption, Alice encrypts a plaintext M for Bob using Bob's public key (n, e) by computing the ciphertext

$$C \equiv M^e \pmod{n}, \tag{1}$$

where n, the modulus, is the product of two or more large primes, and e, the public exponent, is an (odd) integer $e \geq 3$ that is relatively prime to $\phi(n)$, the order of the multiplicative group $\mathbb{Z}_n^*$ [13].

2.2 RSA Digital Signature

$$s \equiv M^d \pmod{n}, \tag{2}$$

where (n, d) is the signer's RSA private key [6, 10]. The signature is verified by recovering the message M with the signer's RSA public key (n, e):

$$M \equiv s^e \pmod{n}. \tag{3}$$

3 Our Methodology

In this section, we would calculate the upper bound and the lower bound of $\phi(n)$ in RSA scheme. The detail de-

*The preliminary version of this paper appeared in Cryptology ePrint Archive: Report 2012/666 [14].

scribed as below.

Notation:
ℓ: Express lower bound.
u: Express upper bound.
ε: A decimal expansion number (e.g $99/100 = 0.99\cdots$).

3.1 Approaching $\phi(n)$

If n is composite, hence

$$\phi(n) \leq n - \sqrt{n}. \tag{4}$$

Sierpinski [22] mentioned it in 1964. It is known that if Equation (4) is a good upper bound for $\phi(n)$. Is there a good lower bound for $\phi(n)$? This question is also be discussed by a newsgroup dialog between Ray Steiner and Bob Silverman in 1999 [23]. For $n > 30$, the $\phi(n) > n^{2/3}$, Kemdall and Osborn proved it [11]. For $n \geq 3$, the $\phi(n) > \frac{\log 2}{2}\frac{n}{\log n}$ given by Hatalova and Salat [9].

3.1.1 Estimate Upper Bound

Is Equation (4) a good upper bound? In the following, we would estimate a new value that is smaller than previous and optimize.

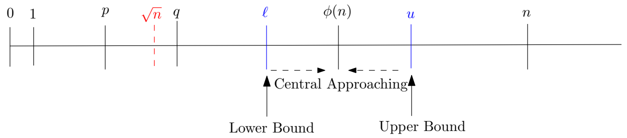


Figure 1: The lower/upper bound of $\phi(n)$ in RSA scheme

Theorem 1. *Assume p and q are large prime numbers, where $n = pq$, then $\phi(n) = 4k$, $k \in \mathbb{Z}$ where $1 \leq k \leq \lfloor\frac{n-2\lceil\sqrt{n}\rceil+1}{4}\rfloor$.*

Proof. As is known, the two variants p and q are large prime numbers. Also both p and $q > 2$, since $2 \nmid p$, $2 \nmid q$, therefore $2 \mid p-1$, $2 \mid q-1$. $4 \mid (p-1)(q-1)$, $4 \mid \phi(n)$. $\phi(n) = 4k$, $k \in \mathbb{Z}^+$. We will discuss how to calculate the range of value k.

$$\begin{aligned} \phi(n) &= (p-1)(q-1) \\ &= pq - (p+q) + 1 \\ &= n - (p+q) + 1. \end{aligned} \tag{5}$$

And

$$\begin{aligned} p+q \geqslant 2\sqrt{n}, p+q \in \mathbb{Z}^+, 2 \mid p+q. \\ p+q \geqslant 2\lceil\sqrt{n}\rceil. \\ \phi(n) \leq n+1-2\lceil\sqrt{n}\rceil. \\ \phi(n) = 4k,\ k \in \mathbb{Z}^+. \\ \phi(n) \leq 4 \cdot \lfloor\frac{n+1-2\lceil\sqrt{n}\rceil}{4}\rfloor. \end{aligned} \tag{6}$$

Here, we know the maximum value (limit superior) for $k \leq \lfloor\frac{n-2\lceil\sqrt{n}\rceil+1}{4}\rfloor$ which we call the boundary value. Consequently, according to the above reference, we obtain an upper bound u of $\phi(n)$ where $\phi(n) \leq 4\lfloor\frac{n-2\lceil\sqrt{n}\rceil+1}{4}\rfloor$. □

3.1.2 Estimate Lower Bound

Loomis *et al.* [15] found Shapiro's [20] lower bound $\phi(n) > n^{(\log 2)}/(\log 3)$ as a (naive) lower bound for E_n, where they can determine when all members of a given E_n have been found. Powell [17] noted that Konyagin's and Shparlinksi's lower bound $N_1(n,p) > (p-1)/2 - p^{3/2}/n$ where $n > 1$ is a positive integer and that p is an odd prime number with $p \equiv 1 \pmod{n}$; it is a good bound if p is a small compared to n, and establishes that

$$N_1(n,p) \geq (\sqrt{\phi(n)}(\prod_{\substack{q\ prime \\ q|n}} q^{1/(q-1)})/n)p^{1-1/\phi(n)}.$$

Powell also discussed an improvement the upper and lower bounds in [17]. What is the optimal lower bound? This is explained as follows.

Theorem 2. *For all $n \geq 3$ we have $\phi(n) \geq \frac{n}{e^{\gamma}\log\log n} + Ø(\frac{n}{(\log\log n)^2})$, where γ is the Euler-Mascherone Constant, and the above holds with equality infinitely often.*
Remark: *note in particular that since loglogn $\to \infty$ as n grows large, we see that the result $\frac{n}{m} < \phi(n)$ can not hold for any fixed integer m.*

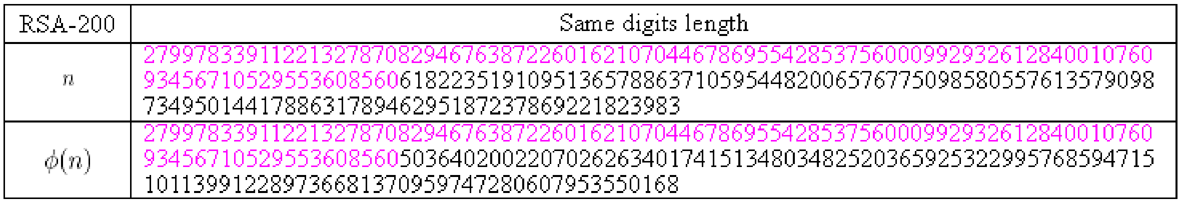
Proof. Consider R, set of all n such that $m < n$ implies $\frac{\phi(n)}{n} < \frac{\phi(m)}{m}$. This set is then all of the 'record breaking' n. If $n \in R$ has k prime factors, let n^* be the product of the first k prime factors. If $n \neq n^*$ and $\frac{\phi(n)^*}{n^*} \leq \frac{\phi(n)}{n}$, which is impossible. Hence, R consist entirely of n of the form $n = \prod_{p \leq y} p$ for some y. Now for $n \in R$, we can choose y so that $\log n = \sum_{p \leq y}\log p = \theta(y)$. Then using one of Mertens estimates we see that $\frac{\phi(n)}{n} = \prod_{p \leq y}(1 - \frac{1}{p}) = \frac{e^{-\gamma}}{\log y} + O(\frac{1}{(\log y)^2})$. Since $\log\log n = \log(\theta(y)) = \log y + Ø(1)$ by Mertens estimates again, we have for $n \in R$, $\phi(n) = \frac{ne^{-r}}{\log\log n} + O(\frac{1}{(\log\log n)^2}$. □

Is there a simple computation method? We observed the modulus n with $\phi(n)$, there are some characteristics. Aa an example for RSA-200, the modulus n and the $\phi(n)$ are 200 decimal digits. We compared n and $\phi(n)$ with each other and found that the first **110** digits are the same. The example is shown in Table 1. A discussion on RSA modulus number with half of the bit prescribed, is introduced in some literatures in [8, 16, 21].

In RSA-704, the n and $\phi(n)$ had same digits 106, it amounts same length with p or q. We computed the upper bound value according to Theorem 1. This upper bound had the same 108 digits with its $\phi(n)$. When we analyzed the RSA-768, the n had 115 digits. The same 115 digits was found with $\phi(n)$; the $\phi(n)$ had the same 120 digits with its upper bound u. See Table 2.

We observed the relationship of $\phi(n)$ and its boundary value k. When $\phi(n)$ is divided by k, we found that the

Table 1: The same digits of $\phi(n)$ and modulus n parameters in RSA



RSA-200	Same digits length
n	27997833911221327870829467638722601621070446786955428537560009929326128400107609345671052955360856061822351910951365788637105954482006576775098580557613579098734950144178863178946295187237869221823983
$\phi(n)$	27997833911221327870829467638722601621070446786955428537560009929326128400107609345671052955360856050364020022070262634017415134803482520365925322995768594715101139912289736681370959747280607953550168

Table 2: Comparison of some types in RSA parameters. Unit: decimal digits

length type	n	$\phi(n)$	$p,\ q$	$n\&\phi(n)$	$\phi(n)\&u$
RSA-200	200	200	100	110	101
RSA-210	210	210	105	?	?
RSA-704	212	212	106	106	108
RSA-220	220	220	109	?	?
RSA-768	232	232	116	115	120

quotient approaches and that, these lower bounders are very close to multiples of number 4. As an example, we say $3.\overline{999}$, and have 106 9's after the decimal point for case of RSA-200 type. The lower bound approximation figure diagram is shown in Figure 2 and in Table 3.

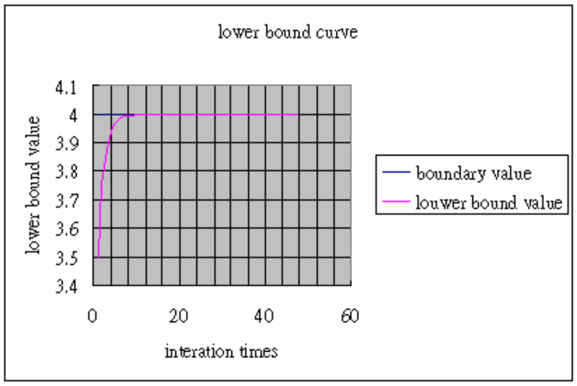


Figure 2: The lower bound approximation curve status.

As known as the modulus number n of RSA-210, we re-estimated its lower/upper bounds. We assume:

$$(3+\varepsilon)\lfloor\frac{n-2\lceil\sqrt{n}\ \rceil+1}{4}\rfloor \le \phi(n) \le 4\lfloor\frac{n-2\lceil\sqrt{n}\ \rceil+1}{4}\rfloor, \qquad (7)$$

where $\varepsilon = 0.\overbrace{99999}^{106's\ 9}$. We therefore compute the upper bound u and lower bound ℓ; those results are shown in Figure 2.

According to Equation (7), the author estimates the upper bound of RSA-220. There are same 109 digits between RSA-220 modulus n and upper bound u. The result is shown in Figure 4.

Table 3: The relationship of $\phi(n)$ and its boundary value k.

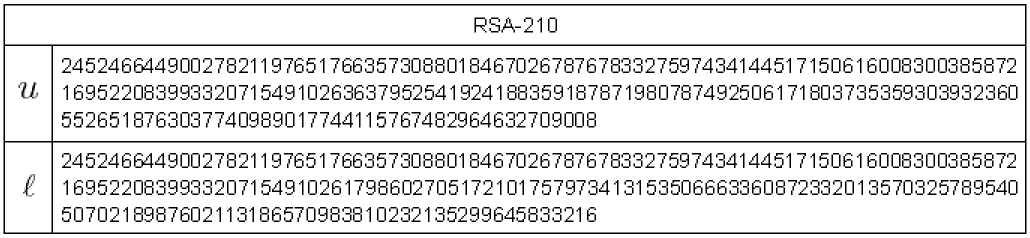
Type	$\phi(n)/k$	Statement
RSA-200	$3.\overbrace{99999}^{99's\ 9}8$	there have 99's 9 after the decimal point
RSA-210	$3.\overbrace{99999}^{106's\ 9}2$	Estimating have 106's 9 after decimal point
RSA-704	$3.\overbrace{99999}^{107's\ 9}8$	there have 107's 9 after decimal point
RSA-220	$3.\overbrace{99999}^{109's\ 9}8$	there have 110's 9 after decimal point
RSA-768	$3.\overbrace{99999}^{117's\ 9}7$	there have 117's 9 after decimal point

4 Conclusion

In this paper, we use another method to estimate a lower/upper bound values of $\phi(n)$ in RSA-210 and upper bound of RSA-220's $\phi(n)$. We think our methodology is easy and intuitive. It may prove useful to researchers who would like to quickly reduce the searching ranges. More researchers focus on secret d or modulus n, such as well known attacks such as short exponent, side channel (or common modulus) and cyclic. Our method is different than previous methods. Finally, the author provides a general method to estimate the lower/upper bound of RSA's $\phi(n)$ public key cryptography.

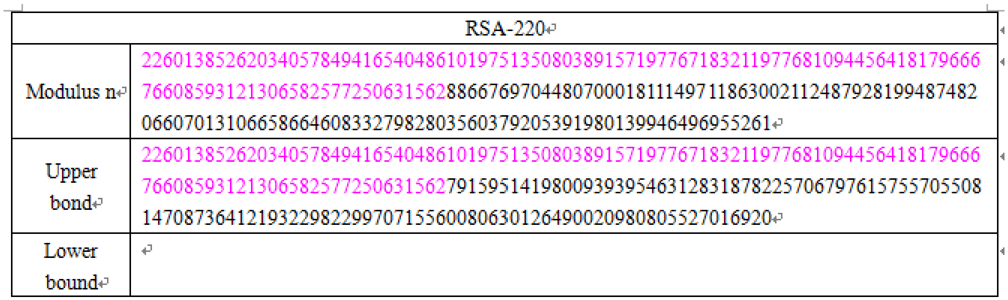
Acknowledgement

The authors would like to thank the anonymous reviewers for their useful comments. This work is partially supported from Huizhou University project under the number HZUXL201513 and HZUXL201514. This work also partially supported by student innovation training program under the grant number CX2016024, CX2016088 and CX2016089. This work is partially supported from Fujian Provincial Department of Education, the education and scientific research projects for young and middle-



RSA-210	
u	24524664490027821197651766357308801846702678767833275974341445171506160083003858721695220839933207154910263637952541924188359187871980787492506171803735359303932360552651876303774098901774411576748296463270900 8
ℓ	245246644900278211976517663573088018467026787678332759743414451715061600830038587216952208399332071549102617986027051721017579734131535066633608723320135703257895405070218987602113186570983810232135299645833216

Figure 3: The lower/upper bound parameters of $\phi(n)$ in RSA-210.



RSA-220	
Modulus n	22601385262034057849416540486101975135080389157197767183211977681094456418179666766085931213065825772506315628866769704480700018111497118630021124879281994874820660701310665866460833279828035603792053919801399464969552 61
Upper bond	22601385262034057849416540486101975135080389157197767183211977681094456418179666766085931213065825772506315627915951419800939395463128318782257067976157557055081470873641219322982299707155600806301264900209808055270169 20
Lower bound	

Figure 4: The upper bound parameters of $\phi(n)$ in RSA-220.

aged teachers where the project number JAT170673. This research partially supported from foundation for high level recruited talents of university; the project number is NUIT2017-001.

References

[1] S. Bai, P. Gaudry, A. Kruppa, E. Thomé, and P. Zimmermann, "Factorisation of rsa-220 with cado-nfs," 2016. (https://en.wikipedia.org/wiki/RSA_numbers#cite_note-33)

[2] S. Bai, E. Thomé, and P. Zimmermann, "Factorisation of RSA-704 with cado-NFS," *IACR Cryptology ePrint Archive*, pp. 369, 2012.

[3] D. Boneh, "Twenty years of attacks on the RSA cryptosystem," *Notices of the AMS*, vol. 46, pp. 203–213, 1999.

[4] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$," *IEEE Transactions on Information Theory*, vol. 46, pp. 1339–1349, July 2000.

[5] C. C. Chang, C. Y. Sun, and S. C. Chang, "A strong RSA-based and certificateless-based signature scheme," *International Journal of Network Security*, vol. 18, no. 2, pp. 201-208, 2016.

[6] L. Deng, H. Huang, and Y. Qu, "Identity based proxy signature from RSA without pairings," *International Journal of Network Security*, vol. 19, no. 2, pp. 229-235, 2017.

[7] J. Franke F. Bahr, M. Boehm and T. Kleinjung, "RSA-200 is factored," 2005. (http://www.rsa.com/rsalabs/node.asp?id=2879)

[8] S. W. Graham and I. E. Shparlinski, "On RSA moduli with almost half of the bits prescribed," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3150–3154, 2008.

[9] H. Hatalova and T. Salat, "Remarks on two results in the elementary theory of numbers," *Acta. FacRer. Natur Univ Comenian. Math.*, no. 20, pp. 113–117, 1969.

[10] M. S. Hwang, K. F. Hwang, I. C. Lin, "Cryptanalysis of the batch verifying multiple RSA digital signatures", *Informatica*, vol. 11, no. 1, pp. 15-19, Jan. 2000.

[11] D. G. Kendall and H. B. Osborn, *Two Simple Lower Bounds for Euler's Function*, vol. 17, Texas Journal of Science, 1965.

[12] T. Kleinjung, K. Aoki, J. Franke, *et al.*, "Factorization of a 768-bit RSA modulus," in *Advances in Cryptology (CRYPTO'10)*, pp. 333–350, Springer, 2010.

[13] C. Liu, C. C. Chang, Z. P. Wu, S. L. Ye, "A study of relationship between RSA public key cryptosystem and Goldbach's conjecture properties," *International Journal of Network Security*, vol. 17, no. 4, pp. 445-453, 2015.

[14] C. Liu and Z. Ye, "Estimating the $\phi(n)$ of Upper/Lower Bound in Its RSA Cryptosystem,". Cryptology ePrint Archive, Report 2012/666, 2012. (http://eprint.iacr.org/2012/666)

[15] P. Loomis, M. Plytage, and J. Polhill, "Summing up the Euler ϕ function," *The College Mathematics Journal*, vol. 39, no. 1, pp. 34–42, 2008.

[16] X. Meng, "On RSA moduli with half of the bits prescribed," *Journal of Number Theory*, vol. 133, no. 1, pp. 105–109, 2013.

[17] C. Powell, "Bounds for multiplicative cosets over fields of prime order," *Mathematics of Computation*, vol. 66, pp. 807–822, Apr. 1997.

[18] R. Propper, "RSA-210 factored," 2013. (http://www.mersenneforum.org/showpost.php?p=354259)

[19] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, 1978.

[20] H. Shapiro, "An arithmetic function arising from the ϕ function," *The American Mathematical Monthly*, vol. 50, pp. 18–30, 1943.

[21] I. E. Shparlinski, "On RSA moduli with prescribed bit patterns," *Designs, Codes and Cryptography*, vol. 39, pp. 113–122, 2006.

[22] W. F. Sierpinski, *Elementary Theory of Numbers*, Warsawa: North-Holland PWN-Polish Scientific Publishers, 1964.

[23] B. Silverman, "Euler's phi functions," 2012. (http://www.math.niu.edu/~rusin/known-math/99/min_phi)

[24] M. J. Wiener, "Cryptanalysis of short RSA secret exponents," *IEEE Transations on Information Theory*, vol. 36, pp. 53–558, Sept. 1990.

Biography

Jie Fang received his associate B.S degree in computer science from Fuqing Branch of Fujian Normal University in 2002 and obtained B.S degree from Fujian Normal University in 2006, and the MSc degree of software engineering at University of Electronic Science and Technology of China in 2010. He is currently a lecturer with school of electronics and information engineering at Fuqing Branch of Fujian Normal University. His research includes Computer Security, E-Learning and Laboratory Management.

Chenglian Liu received his B.S degree in information management from National Union University in 1992 and the MSc degree in National Defense from National Defense University in 2004. He studied his doctorate course at Royal Holloway, University of London from 2006 to 2009 under the supervised by Chris Mitchell. His research interests are in Key Agreement and Password Authentication, Number Theory and Cryptanalysis so on.

Secure and Efficient Authentication Protocol for Power System Computer Networks

Celia Li, Helen Cheung and Cungang Yang
(Corresponding author: Cungang Yang)

Department of Electrical and Computer Engineering, Ryerson University[1]
350 Victoria St, Toronto, ON M5B 2K3, Canada
(Email: cungang@ee.ryerson.ca)

(Received Oct. 16, 2016; revised and accepted Feb. 21 & June 5, 2017)

Abstract

In this paper, we propose an efficient and secure Authentication Protocol for Power systems (APP). The security analysis shows that APP is secure and resilient to various kinds of attacks. The numerical analysis and simulation results shows that APP is more efficient than TLS, a public-key based authentication protocol recommended by IEC61850.

Keywords: Authentication; Network Security; Power System Computer Networks

1 Introduction

Authentication is essential in any service-oriented communication networks to identify and reject any unauthorized network access. An authentication protocol for the power systems must ensure full security to protect data integrity. In addition, the authentication protocol should meet the following requirements from the network perspective [22].

1) High efficiency. Efficiency is crucial to achieve the high availability requirement in real-time power system applications. The indication of high efficiency is two fold. First, the authentication schemes should not incur too much redundancy for security. Second, computation involved in authentication must be fast enough to meet timing requirements of messages in the power systems.

2) Resilient to attacks. Authentication schemes are required to resist malicious attacks, such as forgery attack, replay attack, and DoS attack. In addition, it should be a mutual authentication protocol. Mutual authentication is a two-way authentication process between a user and the authentication server.

The user ensures that he/she is not communicating with a malicious authentication server by authenticating the server. If this property is absent, a malicious authentication server may be able to mount a man-in-the-middle attack to gather private messages from the user. The authentication server also need to authenticate the client to ensure that he/she is communicating with a valid user. The authentication server ensures that he/she is not communicating with a malicious client by authenticating the client. If this property is absent, a malicious user is able to access the network without authentication. Many authentication protocols have been proposed for wireline [20, 27], *e.g.* the Internet, and other types of wireless networks [4, 7, 10, 11, 13, 19, 28, 29]. However, there are few authentication protocol designed for power systems.

Some authentication protocols have been proposed for smart grids [5, 15, 16, 18, 23], but most of them are for meter authentication. Due to the resource constraint of smart meters, the proposed protocols are lightweight (the protocols are based on symmetric key cryptography), which are efficient but not secure. Our authentication protocol is designed for employees of an power site to access sensitive operations or resource of the power system, which needs higher level of security. Our propose protocol is therefore based on public key cryptography. IEC61850 [9] is recently standardized for modern power substation automation by the International Electrotechnical Commission. IEC61850 recommends TLS [2, 11], a public-key based authentication protocol, to achieve secure communications. However, TLS has two weaknesses (1) not efficient; (2) key updates are vulnerable. Therefore, we proposed a new public key-based authentication protocol. Security analysis shows that our protocol is resilient to attacks. Performance analysis demonstrates that our protocol is more efficient than TLS.

2 Related Work

Many authentication protocol have been proposed for wired network, such as the Internet. Kerberos [27] is a computer network authentication protocol which works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Its designers aimed it primarily at a client-server model and it provides mutual authentication-both the user and the server verify each other's identity. Kerberos protocol messages are protected against eavesdropping and replay attacks. Kerberos builds on symmetric key cryptography and requires a trusted third party, and optionally may use public-key cryptography during certain phases of authentication.

The RSA Secure ID [20] employs hardware tokens to authenticate user. The hardware token stores secrets in a tamper-resistant module carried by the user. Here we refer to the simplest dedicated-hardware version, which has only a display and no buttons. Each instance of the device holds a secret "seed" known to the back-end. A cryptographically strong transform generates a new 6-digit code from this secret every 60 seconds. The current code is shown on the device's display. On enrollment, the user connects to the administrative back-end through a web interface, where he selects a PIN and where the pairing between username and token is confirmed. From then on, for authenticating, instead of username and password, the user shall type username and "passcode" (concatenation of a static 4-digit PIN and the dynamic 6-digit code). Many authentication protocols are also introduced for wireless networks. A Protocol for Carrying Authentication for Network Access (PANA) [9], enables authentication between clients and access networks in Wireless Local Area Networks. PANA runs between a client and a server in order to perform authentication and authorization for the network access service. PANA does not define any new authentication mechanisms, but performs authentication protocols of 802.11 standard.

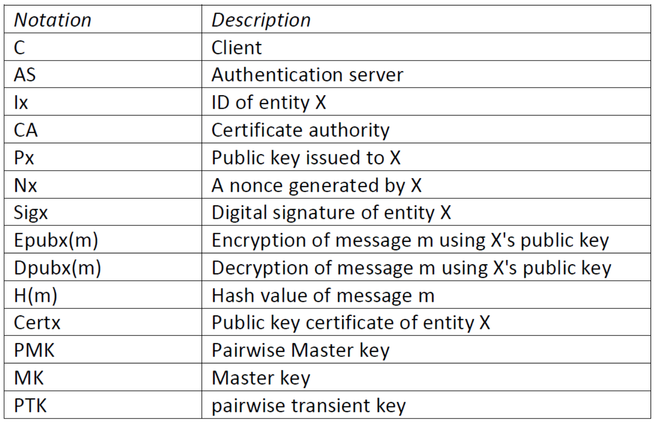
In cellular networks, assume a client roams from a home network to a foreign network, the client needs to be authenticated by the foreign network. The foreign network must communicate with the client's home network via multi-hop communications to authenticate the client [6, 8, 11, 12]. The SIM card of a client and the authentication center of the client's home network are pre-installed with a shared secret key K. When the client roams to a foreign network, the foreign agent must communicate with the client's home network in order to obtain the shared key K, which will then be used to authenticate the client. In the handover authentication protocol of IEEE802.11i standard, after the authentication server successfully authenticates a mobile client, it will send a key called pairwise master key (PMK) to the AP associated with the client. The client will perform the same calculation as the AS to obtain the same PMK. The AP and client will use the PMK to derive a pairwise transient key (PTK) for encrypting future packets exchanged between them [10]. The AS then sends the PMK to the neighbors of the current AP, one by one. The PMK serves as proof of the client's successful login authentication performed by the AS. By letting the AS pre-distribute the PMK to the neighbors of the current AP, the client will not need to be authenticated by the AS when it moves to another AP.

Some authentication protocols for smart grids have been proposed. M. Fouda [5] proposed a light-weight and secure message authentication mechanism. The proposed mechanism is based on Diffie-Hellman key establishment protocol and hash-based message authentication code, which allows various smart meters at different points of the smart grids to make mutual authentication and achieve message authentication with low latency and few signal message exchanges. Li [15] proposed an efficient authentication scheme that employs the Merkle hash tree technique to secure smart gird communication. Specifically, the proposed protocol considers the smart meters with computation-constrained resources and puts the minimum computation overhead on them.

3 The Proposed Authentication Protocol

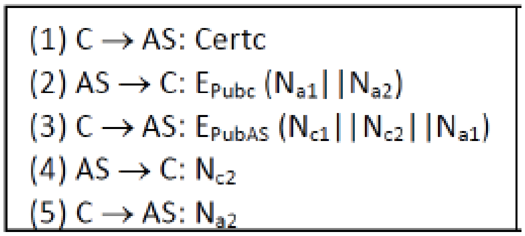
We describe in detail the proposed Authentication Protocol for Power systems (APP). Refer to Table 1 for the notation used in the remainder of the article. Our authentication protocol APP follows a key hierarchical structure similar to that in TLS [25]. That is, a pairwise master key (PMK) is created during the authentication process, and a master key (MK) and pairwise transient key (PTK). The two parties involved in the authentication will used the PTK for point-to-point communications. To minimize the latency of the authentication protocol, the proposed authentication protocol APP aims to minimize (1) the number of message exchanges between a user and the authentication server, thus minimizing communication cost the authentication latency; (2) the number of public key operations performed by the user and the authentication server, thus minimizing computation cost. Here, we assume that all users know the authentication server's public key.

Table 1: Notations



Notation	*Description*
C	Client
AS	Authentication server
Ix	ID of entity X
CA	Certificate authority
Px	Public key issued to X
Nx	A nonce generated by X
Sigx	Digital signature of entity X
Epubx(m)	Encryption of message m using X's public key
Dpubx(m)	Decryption of message m using X's public key
H(m)	Hash value of message m
Certx	Public key certificate of entity X
PMK	Pairwise Master key
MK	Master key
PTK	pairwise transient key

Following are the order of the messages to be exchanged in the protocol and explanation (see Figure 1):



(1) C → AS: Certc
(2) AS → C: E_{Pubc} (N_{a1}| |N_{a2})
(3) C → AS: E_{PubAS} (N_{c1}| |N_{c2}| |N_{a1})
(4) AS → C: N_{c2}
(5) C → AS: N_{a2}

Figure 1: The proposed authentication protocol

1) A client C sends AS a message which contains its public key certificate to inform the AS of its presence and public key. AS verifies the digital signature of the CA who issued the certificate using CA's public key (We assume that AS has the public key certificate of the CA.) AS also verifies other information in the certificate such as the ID of the certificate and the certificate expiry date.

2) If the above verifications are successful, AS extracts the client's public key from client C's certificate $Cert_c$ and generates a message which contains two nonces Na1 and Na2. AS then encrypts the message using C's public key and sends the encrypted message $E_{Pub_{AS}}$ $(Nc1||Nc2)$ to the client C where the operator || denotes a concatenation. Upon receiving the message, C decrypts it using its private key,

3) Client C retrieves AS's public key from AS's public key certificate $Cert_A$ (We assume that client C has the public key certificate of AS and generates two nonces Nc1 and Nc2. C then encrypts Nc1, Nc2 and Na1 using AS's public key, and sends the encrypted message to AS. AS will decrypt the message using its private key to Nc1 and Nc2. Both the client and the AS then calculate a pre-master key $PMK = Nc1||Na1$, where the operator || denotes a concatenation, and Nc1 and Na1 are the nonces generated in Steps 2 and 3 above. (The security of nonces Nc1 and Na1, and thus key PMK is ensured by AS's and client's public-private keys.)

4) AS then sends Nc2 to client C. Upon receiving this message, the client C has successfully authenticated AS, because only AS has the knowledge of Nc2.

5) To allow AS to authenticate C, C sends Na2 (generated by AS in Step 2) to AS. After AS receives Na2 correctly, it is considered to have successfully authenticated client C because only C has the knowledge of Na2.

The AS sends the pre-master key and the random numbers to the web browser, then the user and the browser both calculate the MK and PTKs. Although key generation is not part of this paper, it is worth noting that it is involved partially in the authentication protocol. Key management between a client and the browser allows them to derive a shared key to be used after the authentication for secure data exchanges. We follow a similar approach of key generation defined in TLS. That is, right after Step 3 of the authentication procedure, both parties compute a master key MK as follows:

$$MK = H(PMK, "mastersecret", Na1 + Nc1). \quad (1)$$

After the generation of MK, the two parties use the master key MK to compute a shared key called pairwise transient key (PTK). The PTK will be used to encrypt packets exchanged between the user and the browser.

4 Security Analysis

In this section, we describe the countermeasures implemented in APP against the attacks listed in [3] that are relevant to our protocol.

4.1 Forgery Attack

Forgery attack is an attack in which an attacker deliberately manipulate data. We prevent this type of attack by using digital signatures and message encryption. The public key certificate in the first message uses digital signature to prevent forgery attacks. The digital signature ensures that user C's certificate is protected against modifications and that counterfeit messages are infeasible to be fabricated. Any unauthorized changes to the content of the certificate will result in an incorrect signature value because the attacker does not know CA's private key.

The second and third messages use message encryption to prevent forgery attacks. The encrypted messages are protected against modifications. Any changes to the content of the messages will result in the messages unable to be decrypted successfully by the recipient.

4.2 Replay Attack

An attacker records messages of an ongoing authentication session and replays these messages in the future in an attempt to be successfully authenticated and possibly gain access to the network as a client. An attacker may replay a client's messages to gain access to the network, or an authentication server messages in order to impersonate the server. There are three approaches to resist replay attack. they are nonce, sequence number and time stamps. We prevent this type of attack by using nonces [26]. A nonce is a random number that only can be used for one time. A new message with nonces intended for a specific recipient must use newly generated nonces and not those previously sent to the recipient. If a message with nonces was lost or damaged and the message is retransmitted, the retransmitted message must use newly generated nonces.

We consider possible replay attacks on messages generated by APP described in Section 3.

1) Replaying User Messages.
In the authentication protocol, an attacker overhears and replays Message 1, 3 and 5 sent earlier by a client

C. After successfully receives Message 1 from the attacker, the AS replies with a Message 2. New nonces Na1 and Na2 are generated in the message. The attacker will not be able to decrypt Message 2 because he does not know the private key of AS. The attacker then replays Message 3 to the AS. The AS can detect that this is a replayed message because a new Message 3 is supposed to have new nonce Na1.

2) Replaying AS messages.
In the authentication protocol, an attacker overhears and replays Messages 2 and 4 sent earlier by a AS. The client sends Message 1 to the MAP. The attacker then responds with Message 2. User C sends Message 3 to the attacker. The New nonces Nc1, Nc2 and Na1 are generated in the message. The attacker will not be able to decrypt Message 3 because he does not know the private key of the user C. The attacker replays Message 4. The user C can detect that this is a replay attack because a new Message 4 is supposed to have new nonce Nc2. The attacker will then cannot be authenticated by the AS.

4.3 Denial-of-Service (DoS) Attack

In a DoS attack, a malicious attacker sends a flood of packets to a AS. The network resources are flooded or misused by an attacker to cease service to a legitimate user. In an authentication protocol, an attacker may send bogus messages or replay past valid messages repeatedly to force a AS to use up its resources on processing a large amount of these DoS attack messages. An attacker may repeatedly send copies of Message 1 to a AS. The AS will interpret the duplicates of this message as the losses of Message 2 it has sent. The AS will stop the authentication procedure after a pre-determined number of failed attempts to save resources. Note that this type of attack can happen to any protocol, and not specifically to authentication. An attacker may sniff valid Messages 3 and 5 from a successful authentication and replay the message repeatedly to the involved AS in order to overwhelm it. The AS can detect that this is a replayed attack because the new Messages 3 and 5 are supposed to have new nonces. If the AS receives the replayed message several times, it can infer that it is under a DoS attack and take appropriate actions to thwart the attack [1, 24, 14].

5 Performance Analysis

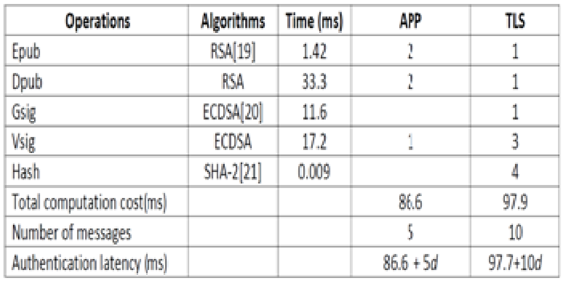
Power communication networks are used to ensure reliable, secure, and real-time message delivery. Hence, latency is much more important than the throughput in power systems, leading to delay-oriented design in power communication protocols. We compare the performance of APP with existing protocol using both numerical analysis and simulations. The protocols to be compared is TLS. TLS is a representative authentication protocol use in power systems recommended by IEC61850. Therefore we chose to compare our protocol with TLS.

5.1 Numerical Analysis

The numerical analysis demonstrates the theoretical gain of our proposed protocols over TLS scheme. The performance of the protocols is measured in terms of Communication costs: which indicate the number of messages exchanged between a AS and a user to complete an authentication session. Computation costs: which are the latencies (in milliseconds) incurred by the following security operations: encryption using public key (E_{pub}); decryption using public key (D_{pub}); generation of a digital signature (G_{sig}); verification of a digital signature (V_{sig}); and hashing. lists the above operations, the current state-of-the-art algorithms implementing the operations, and the computation time each of these algorithms incurs [17] (the first, second and third columns, respectively). The fourth, and fifth columns of Table 2 list the numbers of security operations APP and TLS perform, respectively. By multiplying the computation cost of each operation (from the third column) and the number of times it is executed, and summing up the costs of all operations executed by a protocol, we obtain its total computation cost as shown in the third last row of Table 2.

The computation cost of APP is less than that of TLS. The second last row of Table 2 lists the number of messages exchanged in each protocol. The authentication latencies shown in the last row are the sums of computation costs and communication delays, where d is the average delay of a one transmission incurred by a message. The delay of APP and TLS is $86.6 + 5dms$ vs. $97.7 + 10dms$. The gain of APP over TLS is due to a reduction in the number of messages exchanged, 5 vs. 10 and a reduction of public key operations of APP.

Table 2: Computation and communication costs



Operations	Algorithms	Time (ms)	APP	TLS
Epub	RSA[19]	1.42	2	1
Dpub	RSA	33.3	2	1
Gsig	ECDSA[20]	11.6		1
Vsig	ECDSA	17.2	1	3
Hash	SHA-2[21]	0.009		4
Total computation cost(ms)			86.6	97.9
Number of messages			5	10
Authentication latency (ms)			86.6 + 5*d*	97.7+10*d*

5.2 Simulation Results

We use QualNet (version 5.2), a commercial software that provides scalable simulations of wireless networks [21], for our experiments. The simulation paprameters for all experiments are illustrated in Table 3. The performance metric is authentication delay (latency), which is measured as the time between a client's transmission of an authentication request to an AS and the receipt of an acceptance confirmation. We conduct two sets of experiments to measure the authentication latency as a function of Number of users: We measure the average authentication latency of the proposed protocol and TLS. We measure the average latency by varying group size from 10 to 60. For each data point in a graph, we ran an experiment 10 times using 10 different random seeds and obtained the average rekeying latency. We also keep track of the maximum authentication delay, the maximum value among all users. Background traffic load: We measure the average authentication latency of APP and TLS in the presence of background traffic. We conducted four sets of experiments:

1) We measured the average authentication latency of APP and TLS as a function of number of users. The 400m x 400m network has one node as AS placed in the center of the square. The number of users varied from 10 to 60.

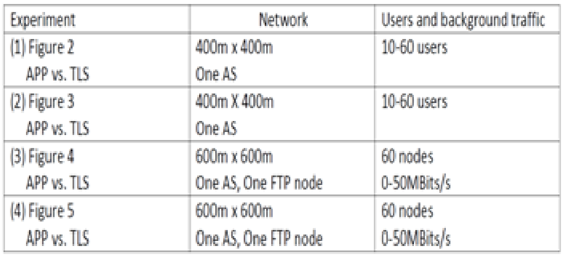
2) We measured the maximum authentication latency of APP and TLS as a function of number of users. We used the same network as in Experiment (1).

3) We measure the average authentication latency of the protocols in the presence of background traffic. The 600m x 600m network has one node as AS placed in the center of the square. We design an additional node as a source to transmit the background traffic of FTP to the AS. This node does not count as a user. The number of users is 60. We vary the data rate of FTP from 0 to 50 Mbits/s in our simulations.

4) We measured the maximum authentication latency of APP and TLS as a function of background traffic. We used the same network as in Experiment (3).

Table 3 summarizes the important parameters and lists the figures containing the graphs of the experiments. In all the experiments, the user nodes were randomly distributed in the networks. To test the scalability of the protocols, we let all users present in the network send authentication requests to the AS simultaneously.

Table 3: Simulation parameters for different experiments



Experiment	Network	Users and background traffic
(1) Figure 2 APP vs. TLS	400m x 400m One AS	10-60 users
(2) Figure 3 APP vs. TLS	400m X 400m One AS	10-60 users
(3) Figure 4 APP vs. TLS	600m x 600m One AS, One FTP node	60 nodes 0-50MBits/s
(4) Figure 5 APP vs. TLS	600m x 600m One AS, One FTP node	60 nodes 0-50MBits/s

5.3 Result Analysis

The results of the above four sets of experiments are illustrated by the graphs in Figures 2, 3, 4 and 5. (1) Figure 2 shows the average authentication latency of the APP vs. TLS under the function of number of users. When there are only 10 users in the network, the average latency of APP and TLS are 167.6ms and 227.2ms respectively. Given more than 10 users, the workload and channel contention at the server further increases. In these cases, the AAP offers lower average latency than TLS, because the APP requires less messages exchanged than EAP-TLS (5 vs. 10, as shown in the second last row of Table 3).

As the number of users increases, the average authentication latency of both APP and TLS increases as well. In the case of 60 users, the average authentication delay of the APP and TLS are 220.1ms and 291.5ms, respectively. The average authentication delay of APP is 24.1% lower than that of TLS. (2) Figure 3 also shows the maximum authentication delay of both protocols. Given 60 users request authentication with the same AS, the maximum authentication delay of APP and TLS are 299.7 ms and 381.6 ms, respectively. The amounts of cryptographic computation performed by LAP and EAP-TLS are very similar (86.6ms vs. 97.7ms as shown in the last row of Table 3). This shows that the gain of APP over TLS is mainly due to their difference on communication costs. (3) We examine how background traffic may affect the average authentication latency and maximum authentication latency if 60 users request to be authenticated at the same time.

Figure 4 shows average authentication latency as function of data rate, which is varied from 10Mbits/s to 50Mbits/s. Data rate is 0 means that there is no background traffic. As the data rate increases, the average authentication latency of users is enlarged. Higher data rate implies more background traffic to be processed by the AS, and more channel contention around the AS, resulting in longer delay. (4) Figure 5 also shows the maximum authentication latency of 60 clients. The data rates varies from 10m/s to 50m/s. As the data rate increases, the maximum authenticate latency of APP and TLS are enlarged. Higher data rate implies more background traffic to be processed by the AS, and more channel contention around the AS, resulting in longer delay.

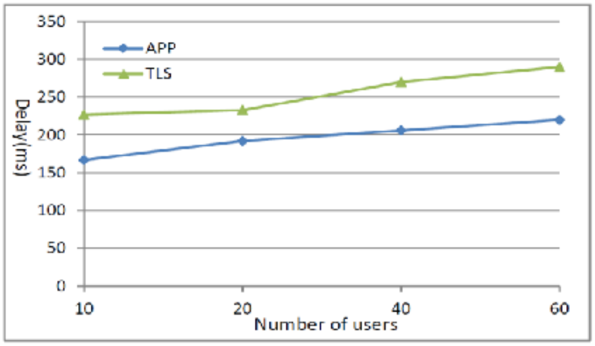


Figure 2: Average latency of APP via TLS - Function of number of users

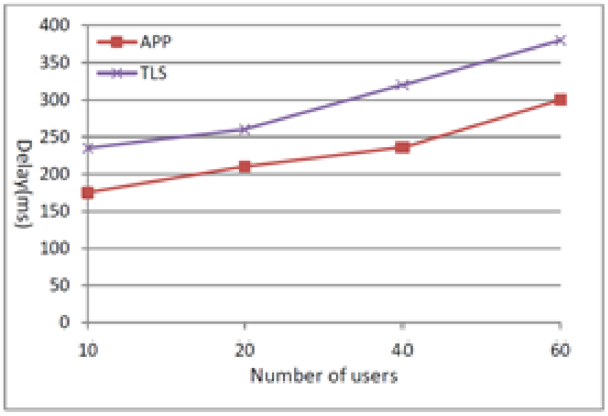


Figure 3: Maximum latency of APP via TLS - Function of number of users

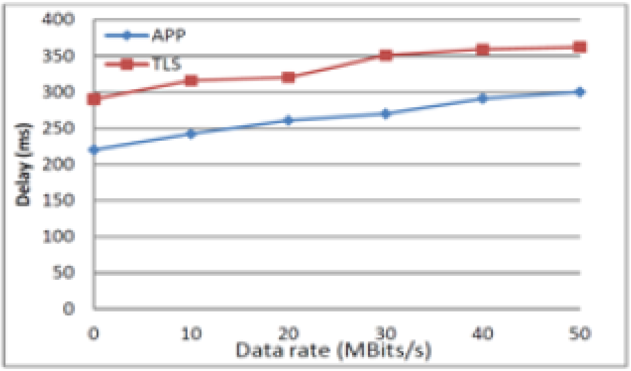


Figure 4: Average latency of APP via TLS - Function of traffic load

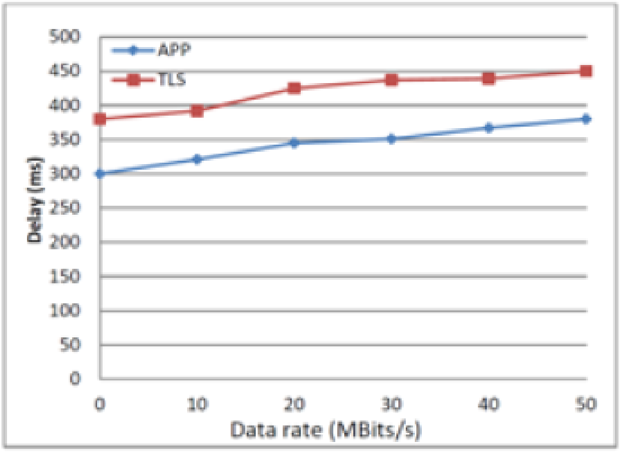


Figure 5: Maximum latency of APP via TLS - Function of traffic load

6 Conclusion

Cyber security in the power systems is a new area of research that has attracted rapidly growing attention in the government, industry and academia. Cyber security is still under development in the power systems, especially because information security must be taken into account with electrical power systems. In this paper, we presented a fast and secure authentication protocol for power systems. The security analysis shows that APP is secure and resilient to various kinds of attacks. The numerical analysis and simulation results shows that APP is more efficient than TLS.

References

[1] T. Aura, P. Nikander, and J. Leiwo, "Dos-resistant authentication with client puzzles," in *The 8th International Workshop Security Protocols*, LNCS 2133, pp. 170–177, Springer, 2000.

[2] K. Bhargavan, C. Fournet, M. Kohlweiss, A. Pironti, P. Strub, and S. Z. B eguelin, "Proving the TLS handshake secure," in *Advances in Cryptology (CRYPTO'14)*, pp. 235-255, Springer, 2014.

[3] FIPS, *Entity Authentication Using Public Key Cryptography*, FIPS Standard PUB 196, 1997.

[4] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, *Protocol for Carrying Authentication and Network Access (PANA)*, Technical Report, RFC 5191, 2008.

[5] M. Fouda, Z. Md. Fadlullah, N. Kato, R. Lu, and X. Shen, "Towards a light-weight message authentication mechanism tailored for smart grid communications," in *IEEE Conference on Computer Communications Workshops*, Apr. 2011.

[6] P. Guo, J. Wang, B. Li, S. Lee, "A variable threshold-value authentication architecture for wireless mesh networks," *Journal of Internet Technology*, vol. 15, no. 6, pp. 929–936, 2014.

[7] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Information Sciences*, vol. 321, pp. 263–277, Nov. 2015.

[8] D. He, S. Zeadally, "Authentication protocol for ambient assisted living system," *IEEE Communications Magazine*, vol. 35, no. 1, pp. 71–77, 2015.

[9] IEC, *Communication Networks and Systems in Substations*, IEC Standard, IEC61850, July 2003.

[10] IEEE, *Part11: Wireless Medium Access Control (MAC) and Physical Layer specifcations: Medium Access Control (MAC) Security Enhancement*, IEEE Standard 802.11i, 2003.

[11] Y. Jiang, C. Lin, X. Shen and M. Shi, "Mutual authentication and key exchange protocols for roaming services in wireless mobile networks," *IEEE Transactionon Wireless Communications*, vol. 5, no. 9, pp. 2569-2577, 2006.

[12] W. I. Khedr, M. I. Abdalla, A. A. Elsheikh, "Enhanced inter-access service network handover authentication scheme for IEEE 802.16 m network," *IET Information Security*, vol. 9, no. 6, pp. 334–343, 2015.

[13] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: A review," *Ad Hoc Networks*, vol. 27, pp. 159-194, Apr. 2015.

[14] J. Lemon, "Resisting SYN flood DoS attacks with a SYN cache," in *Proceedings of the BSD Conference (BSDCON'02)*, pp. 10, 2002.

[15] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient merkle-tree-based authentication scheme for smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 655–663, 2014.

[16] H. Li, *et al.*, "EPPDR: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 20532064, Aug. 2014.

[17] M. Long, "Energy-efficient and intrusion resilient authentication for ubiquitous access to factory floor information," *IEEE Transaction on Industrial Informatics*, vol. 2, no. 1, pp. 40–47, 2006.

[18] K. Mahmood, S. A. Chaudhry, H. Naqvi, T. Shon, and H. F. Ahmad, "A lightweight message authentication scheme for smart grid communications in power sector," *Computers Electrical and Engineering*, vol. 52, pp. 114-124, 2016.

[19] T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the internet of things," *Ad Hoc Networks*, vol. 32, pp. 17-31, Sep. 2015.

[20] RSA, "Two-factor Authentication: RSA SecurID Software Token", 2011. (`https://www.rsa.com/en-us/products/rsa-securid-suite/rsa-securid-access/securid-software-tokens`)

[21] Scalable Networks, *QualNet Simulator*, Dec. 17, 2017. (`http://www.scalablenetworks.com/`)

[22] The Smart Grid Interoperability Panel Cyber Security Working Group, *Guidelines for Smart Grid Cyber Security*, Technical Report, NISTIR 7628, Aug. 2010.

[23] J. L. Tsai and N. W. Lo, "Secure anonymous key distribution scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 906914, 2016.

[24] X. Wang and M. K. Reiter, "Defending against denial-of-service attacks with puzzle auctions (extended abstract)," in *IEEE Symposium on Security and Privacy*, 2003.

[25] Wikipedia, *Transport Layer Security*, DEc. 17, 2017. (`https://en.wikipedia.org/wiki/Transport\_Layer\_Security`)

[26] Wikipedia, *Cryptographic Monce*, Dec. 17, 2017. (`ttp://en.wikipedia.org/wiki/Cryptographic\_nonce`)

[27] Wikipedia, *Kerberos (Protocol)*, Dec. 17, 2017. (`http//:en.wikipedia.org/wiki/Kerberos\_(protocol)`)

[28] H. Xiong, "Cost-effective scalable and anonymous certificateless remote authentication protocol," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2327-2339, Dec. 2014.

[29] H. Xiong and Z. Qin, "Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1442-1455, July 2015.

Appendix: TLS Authentication Protocol

The following full example shows a client being authenticated in addition to the server like above via TLS using certificate exchanged between both peers (see Figure 6).

1) Negotiation Phase: A client sends a ClientHello message specifying the highest TLS protocol version it supports, a random number, a list of suggested cipher suites and compression methods.

   The server responds with a Server Hello message, containing the chosen protocol version, a random number, cipher suite and compression method from the choices offered by the client. The server may also send a session id as part of the message to perform a resumed handshake.

   The server sends its Certificate message (depending on the selected cipher suite, this may be omitted by the server).

   The server sends its Server Key Exchange message (depending on the selected cipher suite, this may be omitted by the server). This message is sent for all DHE and DH anon cipher suites.

   The server requests a certificate from the client, so that the connection can be mutually authenticated, using a Certificate Request message.

   The server sends a Server Hello Done message, indicating it is done with handshake negotiation.

   The client responds with a Certificate message, which contains the client's certificate.

   The client sends a Client Key Exchange message, which may contain a Pre Master Secret, public key, or nothing. (Again, this depends on the selected cipher.) This Pre Master Secret is encrypted using the public key of the server certificate.

   The client sends a Certificate Verify message, which is a signature over the previous handshake messages using the client's certificate's private key. This signature can be verified by using the client's certificate's public key. This lets the server know that the client has access to the private key of the certificate and thus owns the certificate.

The client and server then use the random numbers and Pre-Master Secret to compute a common secret, called the "master secret". All other key data for this connection is derived from this master secret (and the client- and server-generated random values), which is passed through a carefully designed pseudo random function.

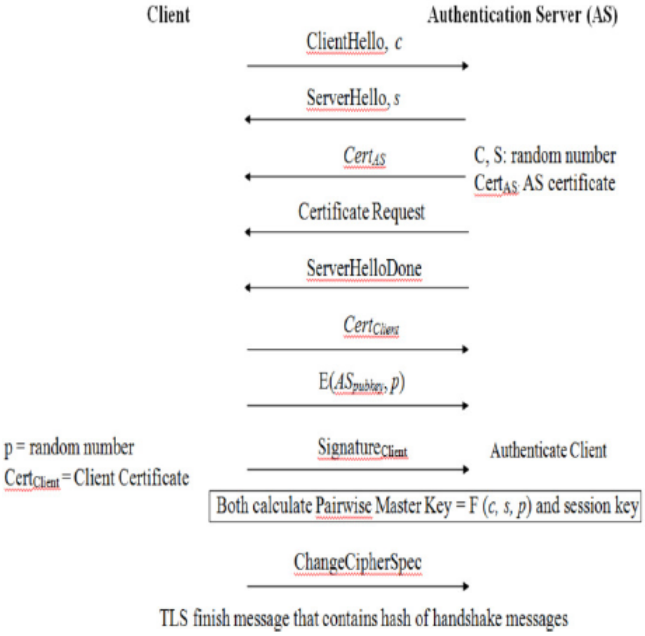

Client
Authentication Server (AS)
ClientHello, c
ServerHello, s
$Cert_{AS}$
C, S: random number
$Cert_{AS}$: AS certificate
Certificate Request
ServerHelloDone
$Cert_{Client}$
$E(AS_{pubkey}, p)$
p = random number
$Cert_{Client}$ = Client Certificate
$Signature_{Client}$
Authenticate Client
Both calculate Pairwise Master Key = F (c, s, p) and session key
ChangeCipherSpec
TLS finish message that contains hash of handshake messages

Figure 6: TLS Authentication Protocol

2) The client now sends a Change CipherSpec record, essentially telling the server, "Everything I tell you from now on will be authenticated (and encrypted if encryption was negotiated). Finally, the client sends an encrypted Finished message, containing a hash and MAC over the previous handshake messages. The server will attempt to decrypt the client's Finished message and verify the hash and MAC. If the decryption or verification fails, the handshake is considered to have failed and the connection should be torn down.

3) Finally, the server sends a Change CipherSpec, telling the client, Everything I tell you from now on will be authenticated (and encrypted if encryption was negotiated). The server sends its own encrypted Finished message. The client performs the same decryption and verification. At this point, the "handshake" is complete

Biography

Celia Li completed her Ph.D degree in electrical engineering and computer science department in 2015 at York University. Her research is focused on security and privacy, role-based access control and wireless mesh network security.

Helen Cheung completed her Ph.D student in electrical and computer engineering department at Ryerson University. Her research interest is on role-based access control, privacy and power system computer network security.

Cungang Yang completed his Ph.D degree in computer science in 2003 at University of Regina, Canada. In 2003, he joined the Ryerson University as an assistant professor in the Department of Electrical and Computer Engineering. His research areas include security and privacy, enhanced role-based access control model, information flow control, web security and secure wireless networks.

Adaptively-Secure Authenticated Key Exchange Protocol in Standard Model

Mojahed Mohamed[1,2], Xiaofen Wang[1], and Xiaosong Zhang[1]

(Corresponding author: Mojahed Ismail Mohamed)

School of Computer Science and Engineering, University of Electronic Science and Technology of China[1]
4 Jianshe North Rd 2nd Section, Chenghua Qu, Chengdu Shi, Sichuan Sheng 610051, China
Department of Electronic Engineering, Karary University[2]
(Email: mmmmoj@hotmail.com)

(Received Nov. 17, 2016; revised and accepted Feb. 21, 2017)

Abstract

Design a Secure Authenticated Key Exchange (AKE) protocol is a wide research area. Many works have been done in this field and remain few open problems. Design an AKE-secure without NAXOS approach is remaining as an open problem. NAXOS approach [18] is used to hide the ephemeral secret key from an adversary even if the adversary in somehow may obtain the ephemeral secret key. Using NAXOS approach will cause two main drawbacks, (1) leaking of the static secret key which will be used in computing the exponent of the ephemeral public key. (2) Maximize of using random oracle when applying to the exponent of the ephemeral public key and session key derivation. Another open problem is designing an AKE-secure in the standard model without relying on Pseudo-Random Function with Pairwise-Independent Random Sources. In this paper, we present a general construction for AKE-secure protocol from the projective hash family secures under hard subset membership problem in the standard model. We also give an instantiation of our protocol from DDH with a novel security proof from games sequences tool introduced by [24]. We show the efficiency of our protocol compares to other similar AKE protocol.

Keywords: AKE; Decision Diffie-Hellman Assumption; eCK Model; Hash Proof System; NAXOS' Approach; Smooth Projective Hash Function

1 Introduction

An Authenticated Key Exchange Protocol is a primitive cryptographic notion which enables two parties after exchanging individual messages to agree on a symmetric shared key used later to secure the channel used between them. The authentication problem deals with restraining adversary that actively controls the communication links used by legitimated parties. They may modify and delete messages in transit, and even inject false one or may control the delays of messages.

Bellare nad Rogaway [1] formalized the security of KE protocols in the realistic setting of synchronal sessions running in a network controlled by the adversary. Their work focused on the shared-key model of authentication while other works [2, 3] expand the techniques to include public-key setting.

Canetti-Krawczyk [5] provides Adopted model for [25] with extraction of construction of secure sessions.

LaMacchia, Lauter and Mityagin [18] Presents significant security model for Authenticated Key Exchange (AKE) protocols which it is extending to Canetti-Krawczyk model. This model capture attacks resulting from leakage of ephemeral and long-term secret keys defined by an experiment in which the adversary is given many corruption power for various key exchange sessions and most solve a challenge on a test session. Moreover, This model doesn't give an adversary capability to break an AKE protocol trivially.

Recently a variant of eCK model used in literatures (e.g., [8, 9, 17, 26]). The difference is those models using StateReveal query instead of EphemeralKeyReveal query in the eCK model, which models maximum exposure.

Bresson *et al.* [4] used a secure device together with an untrusted host machine to attain the existing gap between formal models and effective security. A secure device may use to store long-term keys and, at least, be able to perform some mathematical functions (such as addition, modulo, and exponentiation) which are necessary to achieve cryptographic operations. In such way, we could assume that ephemeral keys and intermediate states generated on host machine are liable to StateReveal attacks to model the maximum state leakage attack (MSL). Although there might exist some side-channel attacks (such as [16]) against the secure device, we assume it works as a black-box to avoid the leakage of internal states [26].

When using the secure device then the security model would equal to a model without StateReveal query. However, the security result of a protocol analyzed with such implementation scenario must be weaker than that in a case allowing leakage of states. In contrast, our goal is to define the maximum states that can be leaked. The secure devices have limited resources which may cause performance bottleneck of systems.

In this paper, we will use eCK for several security attributes, as resistance to key compromise impersonation attack (KCI), leakage of secret states and chosen identity and public key attack (CIDPK) and provide of weak perfect forward secrecy (wPFS). Also, we will use StateReveal query instead of EphemeralKeyReveal query to models maximum disclosure.

Motivating Problem

(1) In AKE, still remind few open problems. Is it essentially to use NAXOS trick [18] in designing the AKE protocols. This method is used to hide the ephemeral secret key from an adversary even if the adversary in somehow may obtain the ephemeral secret key. Design AKE-secure protocol without NAXOS trick will achieve two goals: (i) To reduce the risk of leaking the static private key, since the derivation of the ephemeral public key is independent of the static private key. This method in contrast to protocols that use the NAXOS' approach. (ii) Minimize the utilization of the random oracle, by applying it only to the session key derivation. Kim, Minkyu, Atsushi Fujioka, and Berkant Ustaolu [15, 19] proposed a two strongly secure authenticated key exchange protocols without NAXOS approach, one of their protocol supposed to be secure under the GDH assumption and the other under the CDH assumption in random oracle model. Mohamed *et al.* [20] designed a protocol without NAXOS approach but secure in RO model. Recently, Daisuke *et al.* [12] presents an eCK secure AKE protocol without using the NAXOS trick, but they still rely on Pseudo-Random Function with Pairwise-Independent Random Sources. In another hand, we can find several protocols designed with NAXOS trick and supposed to be secure in a different manner of definition. Those protocols should answer the question how to implement the NAXOS trick securely. In the original implementation, the hash function will be used as in original NAXOS protocol [18]. In some design, we can found the exponent of the ephemeral secret key hidden with a particular kind of PRF [9, 22]. In some scenario, secure device may use to cover up the ephemeral secret key. The remain challenges are the limitation of computational capability of those devices and limitation of resources. (2) Design AKE-secure without NAXOS trick in the standard model. As mentioned above that secure device might not be the ideal solution because of its short in storage capacity and computational resources. (3) Design adaptive AKE-secure using a hash proof system. Cramer and Shop [7] invented the universal hash proof system. It is a particular kind of non-interactive zero-knowledge proof system for the language. They show how to construct an efficient public-key encryption schemes secure against adaptive chosen ciphertext attack in the standard model given an efficient universal hash proof system. [10] presented a general framework for password-based authenticated key exchange protocols using modified smooth projective hash function.Katz *et al.* [13] introduced password-based authenticated key exchange protocol. Their protocol uses a CCA-secure labeled public-key encryption scheme (Gen, Enc, Dec), and a smooth projective hash function. That protocol does not consider the attack of StaticKeyReveal, SessionKeyReveal or EstablishParty, which causes static secret keys leakage and session keys leakage. The protocol follows plan security model which not consider the session freshness definition and needed to be shared password with the public keys.

Contributions

In this paper, we present a concrete and practical AKE-secure protocol which is eCK secure under Decisional Diffie-Hellman assumption in the standard model. Our protocol does not rely on NAXOS trick or Pseudo-Random Function with Pairwise-Independent Random Sources as [12, 22]. We give a generic construction for AKE-secure protocol from the projective hash family secure under hard subset membership problem in the standard model. We also provide an instantiation of our protocol from DDH with a novel security proof from games sequences tool introduced by [24]. We show the efficiency of our protocol compares to other similar AKE protocol.

Organization

Section 2 reviews security definitions, general assumptions and states the hard problem. Section 3 gives brief for the eCK model. Section 4 proposes a Generic adaptively-secure AKE Construction from HPS. Section 5 presents an instantiation from the DDH Assumption for paradigm designed in Section 4. And finally we draws the conclusion in Section 6.

2 Preliminaries

Notation

Let [n] denote the set $\{1, ..., n\}$. Let $k \in \mathbb{N}$ denote the security parameter and 1^k denote the string of k ones. $s \leftarrow_{\$} S$ denotes picking an element s uniformly random from S. $y \leftarrow \mathcal{A}(x)$ denotes runnung $\mathcal{A}$ with input x and assigning y as the result. $logs$ denotes logarithm s for base 2. Let $\Delta(X; Y)$ be the *statistical distance* between two random variables X and Y having a common domain $\mathcal{X}$.

2.1 Randomness Extractor

Entropy

is a measurement of unpredictable of information content.

Definition 1 (Entropy)**.** *entropy* $\mathrm{H}(.)$ *of a discrete random variable* X *with possible values* $\{x_1, ..., x_n\}$ *and probability mass function* $Pr[X]$ *defined as:*

$$\mathrm{H}(X) = \mathbb{E}[-ln(Pr[X])] = -\sum Pr[x_i] \log Pr[x_i]$$

Min-entropy

The min-entropy of a distribution X (denoted $\mathrm{H}_\infty(X)$), is the largest real number k such that $\Pr[X = x] \leq 2^{-k}$ for every x in the range of X. In essence, this measures how likely X is to take its most likely value, giving a worst-case bound on how random X appears. Letting U_ℓ denote the uniform distribution over $\{0,1\}^\ell$, clearly $\mathrm{H}_\infty(U_\ell) = \ell$.

For an $n - bit$ distribution X with min-entropy k, we say that X is an (n, k) distribution.

Definition 2 (Randomness Extractor)**.** *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$ *be a function that takes as input a sample from an* (n, k) *distribution* X *and a d-bit seed from* U_d*, and outputs an m-bit string.* Ext *is a* (k, ε)*-extractor, if for all* (n, k) *distributions* X*, the output distribution of Ext is* ε*-close to* U_m*.*

Definition 3 ((k, ϵ)-Strong Extractor)**.** *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$ *is a* (k, ϵ)*-strong extractor such that for any* x *distributed over* X *that has min-entropy* k *and for any seed* $s \leftarrow_{\$} \{0,1\}^d$ *which is chosen uniformly at random from* (d, k) *distribution and for any value* $r \leftarrow_{\$} \{0,1\}^m$ *which is chosen uniformly at random from* (m, k) *distribution, the two distributions* $\langle s, Ext(s, x)\rangle$ *and* $\langle s, r\rangle$ *have statistical distance at most* ϵ

$$\frac{1}{2} \sum_{y \in (m,k)} |Pr[Ext(s, x) = y] - Pr[r = y]| = \epsilon$$

Some good results on key derivation and randomness extraction can be also found in [6].

2.2 Pseudo-Random Functions

A pseudo-random function PRF is a deterministic functions introduced by Goldreich, Goldwasser and Micali [11]. Let $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \rightarrow \mathcal{R}_prf$ denote a family of deterministic functions, where $\mathcal{K}_{\mathsf{PRF}}$ is the key space, $\mathcal{D}_{\mathsf{PRF}}$ is the domain and $\mathcal{R}_prf$ is the range of PRF for security parameter λ. Let $RL = \{(x_1, y_1), ..., (x_q, y_q)\}$ be a list which is used to record bit strings formed as tuple $(x_i, y_i) \in (\mathcal{D}_{\mathsf{PRF}}, \mathcal{R}_prf)$ where $1 \leq i \leq q$ and $q \in \mathbb{N}$. In RL each x is associated with a y. Let $RF : \mathcal{D}_{\mathsf{PRF}} \rightarrow \mathcal{R}_prf$ be a stateful uniform random function, which can be executed at most a polynomial number of q times and keeps a list RL for recording each invocation. On input a message $x \in \mathcal{D}_{\mathsf{PRF}}$, the function $RF(x)$ is executed as follows:

- If $x \in RL$, then return corresponding $y \in RL$.
- Otherwise return $y \leftarrow_{\$} \mathcal{R}_{\mathsf{PRF}}$ and record (x, y) into RL.

Definition 4. *We say that* PRF*is a* $(q, t, \epsilon_{\mathsf{PRF}})$*-secure pseudo-random function family, if it holds that*

$$\left| \Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\lambda) = 1 \right] - \frac{1}{2} \right| \leq \epsilon_{\mathsf{PRF}}$$

for all adversaries $\mathcal{A}$ *that makes a polynomial number of oracle queries* q *while running in time at most* t *in the following experiment:*

```
$EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\lambda)$
$k \leftarrow_{\$} \{0,1\}^\lambda$,
$b \leftarrow_{\$} \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{F}(b,.)}(\lambda)$
return $b = b'$
```

```
$\mathcal{F}(b, x)$
for $i = 1..q, q \in \mathbb{N}$ do
if $x \notin \mathcal{D}_{\mathsf{PRF}}$return $\perp$
$z_0 = \mathsf{PRF}(k, x), z_1 = RF(x)$
return $z_b$
```

2.3 Hash Proof System

Cramer and Shoup [7] introduced a novel security notion called universal hash proof system. They showed that given this system how to construct efficient public-key encryption schemes secure against adaptive chosen ciphertext attack in the standard model. In another hand, we can describe the hash proof system as a key encapsulation mechanism (KEM) [14, 23] with special algebraic properties.

Universal Hashing

Let $\mathcal{SK}, \mathcal{PK}, \mathcal{K}, \mathcal{C}$ and $\mathcal{V}$ be non-empty finite sets, represents secret keys, public keys, encapsulated keys, ciphertext set and valid ciphertext set respectively. Where $\mathcal{V} \subset \mathcal{C}$. Let $\Lambda_{sk} : \mathcal{C} \rightarrow \mathcal{K}$ be an indexed hash function indexed by sk. We call Λ_{sk} *projective* if there exists a projection $\mu : \mathcal{SK} \rightarrow \mathcal{PK}$ where $\mu(sk) \in \mathcal{PK}$ defines the action of Λ_{sk} over the subset $\mathcal{V}$. That is, for every $C \in \mathcal{V}$, the value $K = \Lambda_{sk}(C)$ is uniquely determined by $\mu(sk)$ and C. In contrast, nothing is guaranteed for $C \in \mathcal{C} \setminus \mathcal{V}$, and it may not be possible to compute $\Lambda_{sk}(C)$ from $\mu(sk)$ and C.

Definition 5 (Universal)**.** *A projective hash function* Λ_{sk} *is* ϵ*-universal, if for all* $pk, C \in \mathcal{C} \setminus \mathcal{V}$*, and all* $K \in \mathcal{K}$*, it holds that* $\Pr[\Lambda_{sk}(C) = K | (pk, C)] \leq \epsilon$*, in other word we can say*

$$\Delta[(pk, \Lambda_{sk}(C)); (pk, K)] \leq \epsilon$$

where in the above $pk = \mu(sk)$ *for* $sk \leftarrow_{\$} \mathcal{SK}$ *and* $K \leftarrow_{\$} \mathcal{K}$*.*

Lemma 1. *Assume that* $\Lambda_{sk} : \mathcal{C} \rightarrow \mathcal{K}$ *is an* ϵ*-universal projective hash function. Then, for all* pk *and* $C \in \mathcal{C} \setminus \mathcal{V}$*, it holds that* $H_\infty(\Lambda_{sk}(C) | (pk, C)) \geq \log 1/\epsilon$*, where* $sk \leftarrow \mathcal{SK}$ *with* $pk = \mu(sk)$*.*

Hash Proof System

A hash proof system **HPS = (Gen, Pub, Priv)** consists of three algorithms. The parameter generation algorithm

$HPS.Gen(1^k)$ generates parameterized instances of the form $params = (group, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \Lambda_{(.)} : \mathcal{C} \rightarrow \mathcal{K}, \mu : \mathcal{SK} \rightarrow \mathcal{PK})$, where group may contain additional structural parameters. The public evaluation algorithm $HPS.Pub(pk, C, w)$ takes as input a projective public key $pk = \mu(sk)$, a valid ciphertext $C \in \mathcal{V}$ and a witness w of the fact that $C \in \mathcal{V}$, and computes the encapsulated key $K = \Lambda_{sk}(C)$. The private evaluation algorithm $HPS.Priv(sk, C)$ takes a secret key sk and a ciphertext $C \in \mathcal{V}$ as input, and returns the encapsulated key $K = \Lambda_{sk}(C)$ without knowing a witness. We assume that μ and $\Lambda_{()}$ are efficiently computable.

We say that a hash proof system is universal if for all possible outcomes of $HPS.Gen(1^k)$ the underlying projective hash function is ϵ-almost universal for negligible $\epsilon(k)$. Furthermore, we say that a hash proof system is perfectly universal if $\epsilon(k) = 0$.

Subset Membership Problems

The subset membership problem associated with a HPS suggests that a random valid ciphertext $C_0 \leftarrow_{\$} \mathcal{V}$ and a random invalid ciphertext $C_1 \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}$ are computationally indistinguishable. This is formally captured by a negligible advantage function $\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(k)$ for all PPT adversary $\mathcal{A}$, where

$$\begin{aligned}\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(k) \quad = \quad & |\Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_0) = 1 | C_0 \leftarrow_{\$} \mathcal{V}] \\ & - \Pr[\mathcal{A}(\mathcal{C}, \mathcal{V}, C_1) = 1 | C_1 \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}] \,|.\end{aligned}$$

Definition 6. *A hash proof system* ***HPS = (HPS.Gen, HPS.Pub, HPS.Priv)*** *is ϵ-universal if: (i) for all sufficiently large $k \in \mathbb{N}$ and for all possible outcomes of $HPS.Gen(1^k)$, the underlying projective hash function is $\epsilon(k)$-universal for negligible $\epsilon(k)$; (ii) the underlying subset membership problem is hard. Furthermore, a hash proof system is called perfectly universal if $\epsilon(k) = 1/|\mathcal{K}|$.*

2.4 The DDH Assumption

We assume a PPTalgorithm $\mathcal{G}(1^k)$ that takes as input 1^k and outputs a tuple of $\mathbb{G} = \langle q, G, g \rangle$, where G is a cyclic group of prime order q and g is a generator of G. The Decisional Diffie-Hellman (DDH) assumption holds iff

$$\mathsf{Adv}^{\mathrm{ddh}}_{G,\mathcal{D}}(k) = \\ \left| \Pr[\mathcal{D}(g_1, g_2, g_1^r, g_2^r) = 1] - \Pr\left[\mathcal{D}(g_1, g_2, g_1^r, g_2^{r'}) = 1\right] \right|$$

is negligible in k for any PPT adversary $\mathcal{D}$, where $g_1 \leftarrow_{\$} G$, $g_2 \leftarrow_{\$} G$, $r \leftarrow_{\$} \mathbb{Z}_q$ and $r' \leftarrow_{\$} \mathbb{Z}_q \setminus \{r\}$.

3 Security Model

In this section, eCK model is outlined. In eCK model there are n different parties $P = P_1, , P_n$ running the KE protocol Π. Each party possesses a pair of long-term static (private/public) keys together with a corresponding certificate issued by a certificate authority. The protocol Π is executed between two parties say $\mathcal{A}$ and $\mathcal{B}$, whose static public key are A and B respectively. These two parties exchange their ephemeral public keys X and Y and obtain the same final session key.

Sessions

A party is activated by an outside call or an incoming message to execute the protocol Π. Each program of executing Π is modeled as an interactive probabilistic polynomial-time machine. We call a session an invocation of an instance of Π within a party. We assume that $\mathcal{A}$ is the session initiator and $\mathcal{B}$ is the session responder. Then $\mathcal{A}$ is activated by the outside call $(\mathcal{A}, \mathcal{B})$ or the incoming message $(\mathcal{A}, \mathcal{B}, Y)$. When activated by $(\mathcal{A}, \mathcal{B})$, $\mathcal{A}$ prepares an ephemeral public key X and stores a separate session state which includes all session-specific ephemeral information. The session identifier (denoted by sid) in $\mathcal{A}*$ is initialized with $(\mathcal{A}, \mathcal{B}, X, -, \mathcal{I})$. After $\mathcal{A}$ is activated by $(\mathcal{A}, \mathcal{B}, Y)$ (receiving an appropriate message from responder), the session identifier is updated to $(\mathcal{A}, \mathcal{B}, X, Y, \mathcal{I})$. Similarly, the responder $\mathcal{B}$ is activated by the incoming message $(\mathcal{B}, \mathcal{A}, X)$. When activated, $\mathcal{B}$ also prepares an ephemeral public key Y and stores a separate session state, and the corresponding session identifier is $(\mathcal{B}, \mathcal{A}, Y, X, \mathcal{R})$. A $(\mathcal{B}, \mathcal{A}, Y, X, \mathcal{R})$ (if it exists) is said to be matching to the session $(\mathcal{A}, \mathcal{B}, X, Y, \mathcal{I})$ or $(\mathcal{A}, \mathcal{B}, X, -, \mathcal{I})$. For a session $(\mathcal{A}, \mathcal{B}, *, *, role)$, $\mathcal{A}$ is called the owner of the session while $\mathcal{B}$ is called the peer of the session. We say sid is complete if there is no symbol ”” in sid.

Adversaries

The adversary $\mathcal{M}$ is also modeled as a probabilistic polynomial-time machine. $\mathcal{M}$ controls the whole communications between parties by sending arbitrary messages to the intended party on behalf of another party and receiving the outgoing message from the communicating parties. To capture the possible attacks, $\mathcal{M}$ is allowed to make the following queries.

EstablishParty($\mathcal{U}$): $\mathcal{M}$ Registers an arbitrary party $\mathcal{U}$ not in P, whose static public key is on $\mathcal{M}$s own choice. We call this kind of new registered parties dishonest ($\mathcal{M}$ totally controls the dishonest parties), while the parties in P are honest. We require that when $\mathcal{M}$ makes such query, the certificate authority(CA) should verify that the submitted static public key is in the appropriate group (to avoid small subgroup attack) and the proof that $\mathcal{M}$ knows the corresponding static private key.

Send($\mathcal{A}$,m): $\mathcal{M}$ sends the message m to party $\mathcal{A}$. Upon invocation $\mathcal{A}$ by m, the adversary obtains the outgoing message of $\mathcal{A}$.

StateReveal(sid): $\mathcal{M}$ obtains the secret state stored in the session state of the session sid.

StaticKeyReveal(P_i): $\mathcal{M}$ learns the long-term static private key of an honest party P_i. In this case, P_i no longer seems honest.

SessionKeyReveal(sid): $\mathcal{M}$ obtains the session key for the session sid if the it has been accepted, otherwise $\mathcal{M}$ obtains nothing.

Experiment

$\mathcal{M}$ is given the set P of honest parties, and makes whichever queries he wants. The final aim of the adversary is to distinguish a session key from a random string of the same length. Thus $\mathcal{M}$ selects a complete and fresh session sid, and makes a special query *Test(sid)*. This query can be queried only once, and the session sid is called test session. On this query, a coin b is flipped, if $b = 1$ $\mathcal{M}$ is given the real session key held by sid, otherwise $\mathcal{M}$ is given a random key drawn from the key space at random. $\mathcal{M}$ wins the experiment if he guesses the correct value of b. Of course, $\mathcal{M}$ can continue to make the above queries after the *Test* query; however the test session should remain fresh throughout the whole experiment.

Definition 7 (Fresh session)**.** *Let sid be a complete session, owned by honest $\mathcal{A}$ with honest peer $\mathcal{B}$. If the matching session of sid exists, we let $\overline{sid}$ denote the session identifier of its matching session. sid is said to be fresh if none of the following events occurs:*

1) *$\mathcal{M}$ makes a **SessionKeyReveal(**sid**)** query or a **SessionKeyReveal**($\overline{sid}$) query if $\overline{sid}$ exists.*

2) *If $\overline{sid}$ exists, $\mathcal{M}$ makes either of the following queries:*

   a. *Both **StaticKeyReveal**($\mathcal{A}$) and **StateReveal**(sid), or*

   b. *Both **StaticKeyReveal**($\mathcal{B}$) and **StateReveal**($\overline{sid}$).*

3) *If $\overline{sid}$ does not exist, $\mathcal{M}$ makes either of the following queries:*

   a. *Both **StaticKeyReveal**($\mathcal{A}$) and **StateReveal**(sid), or*

   b. ***StaticKeyReveal**($\mathcal{B}$).*

The eCK security notion can be described now.

Definition 8 (eCK security)**.** *The advantage of the adversary $\mathcal{M}$ in the above experiment with respect to the protocol Π is defined as (b is the guessed value of coin by $\mathcal{M}$):*

$$Adv_{\Pi}^{AKE}(\mathcal{M}) = |2\Pr[b' = b] - 1| \qquad (1)$$

The protocol Π is said to be secure if the following conditions hold:

1) *If two honest parties complete matching sessions, then they will both compute the same session key, except with a negligible probability.*

2) *The advantage of the adversary $\mathcal{M}$ is negligible.*

4 A Generic Adaptively-secure AKE Construction from HPS

In this section, we present a generic authenticated key exchange protocol from HPS.This protocol can be implemented to ensure eCK adaptive security.

4.1 Protocol Description

Parameters

Let $HPS = (HPS.Gen, HPS.Pub, HPS.Priv)$ be an ϵ_{hps}-universal hash proof system, where $HPS.Gen(1^{\lambda})$ generates instances of $params = (group, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \Lambda_{(.)}: \mathcal{C} \to \mathcal{K}, \mu : \mathcal{SK} \to \mathcal{PK})$. Let $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a $(\nu, \epsilon_{\mathsf{Ext}})$-Strong Extractor. Let $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_{\mathsf{PRF}}$ be a $(q, t, \epsilon_{\mathsf{PRF}})$-secure pseudorandom function family. We assume that $\epsilon_{\mathsf{PRF}}, \epsilon_{\mathsf{Ext}}$ and ϵ are negligible λ.

Key Generation

At beginning of the protocol, $HPS.Gen(1^{\lambda})$ will be run for once to generate the public parameter ($param$). A party $\hat{A}$ picks $sk_{\hat{A}} \leftarrow_{\$} \mathcal{SK}$ and sets $pk_{\hat{A}} = \mu(sk)$. The public/secret key for the party $\hat{A}$ is $(sk_{\hat{A}}, pk_{\hat{A}})$. Similarly, a party $\hat{B}$ will set his public/secret keys as $(sk_{\hat{B}}, pk_{\hat{B}})$. We assume this protocol executed between party $\hat{A}$ and party $\hat{B}$, where party $\hat{A}$ is the initiator and party $\hat{B}$ is the responder.

Messages Exchange

The party $\hat{A}$ chooses $C_{\hat{A}} \leftarrow \mathcal{V}$ with witness $\omega_{\hat{A}}$, a random seed $s_{\hat{A}} \leftarrow_{\$} \{0,1\}^d$ and then computes

$$k_{\hat{A}} \leftarrow HPS.Pub(pk_{\hat{B}}, C_{\hat{A}}, \omega_{\hat{A}}), \Phi_{\hat{A}} \leftarrow \mathsf{Ext}(k_{\hat{A}}, s_{\hat{A}})$$

Then sends $(\hat{B}, \hat{A}, C_{\hat{A}}, \Phi_{\hat{A}}, s_{\hat{A}})$ to party $\hat{B}$. Simulatly, the part $\hat{B}$ will follow the same steps and sends $(\hat{A}, \hat{B}, C_{\hat{B}}, \Phi_{\hat{B}}, s_{\hat{B}})$ to party $\hat{A}$.

Upon receiving $(\hat{B}, \hat{A}, C_{\hat{A}}, \Phi_{\hat{A}}, s_{\hat{A}})$, party $\hat{B}$ uses his secret key to get $k'_{\hat{A}} \leftarrow HPS.Priv(sk_{\hat{B}}, C_{\hat{A}})$, then computes $\Phi'_{\hat{A}} \leftarrow \mathsf{Ext}(k'_{\hat{A}}, s_{\hat{A}})$. The party $\hat{B}$ checks $\Phi_{\hat{A}} = \Phi'_{\hat{A}}$, if not then halt. Otherwise, the party $\hat{B}$ computes the session key.

Session Key

The party $\hat{B}$ compute his session key as following:

1) $K \leftarrow k'_{\hat{A}} \oplus k_{\hat{B}}$.

2) $seed \leftarrow \hat{A} \parallel \hat{B} \parallel pk_{\hat{A}} \parallel pk_{\hat{B}} \parallel C_{\hat{A}} \parallel C_{\hat{B}} \parallel k'_{\hat{A}} \parallel k_{\hat{B}}$.

3) $k_s \leftarrow \mathsf{PRF}(K, seed)$.

The correctness of the above protocol follows from the correctness of the underlying hash proof system.

$\hat{A}$		$\hat{B}$
$sk_{\hat{A}} \leftarrow_{\$} \mathcal{SK}; pk_{\hat{A}} \leftarrow \mu(sk)$		$sk_{\hat{B}} \leftarrow_{\$} \mathcal{SK}; pk_{\hat{B}} \leftarrow \mu(sk)$
$C_{\hat{A}} \leftarrow \mathcal{V}$ **with witness** $\omega_{\hat{A}}$		$C_{\hat{B}} \leftarrow \mathcal{V}$ **with witness** $\omega_{\hat{B}}$
$s_{\hat{A}} \leftarrow_{\$} \{0,1\}^d$		$s_{\hat{B}} \leftarrow_{\$} \{0,1\}^d$
$k_{\hat{A}} \leftarrow HPS.Pub(pk_{\hat{B}}, C_{\hat{A}}, \omega_{\hat{A}})$		$k_{\hat{B}} \leftarrow HPS.Pub(pk_{\hat{A}}, C_{\hat{B}}, \omega_{\hat{B}})$
$\Phi_{\hat{A}} \leftarrow \mathsf{Ext}(k_{\hat{A}}, s_{\hat{A}})$		$\Phi_{\hat{B}} \leftarrow \mathsf{Ext}(k_{\hat{B}}, s_{\hat{B}})$
$sid = (\hat{A}, \hat{B}, C_{\hat{A}}, -, \mathcal{I})$		$\overline{sid} = (\hat{B}, \hat{A}, C_{\hat{B}}, -, \mathcal{R})$
	$\xrightarrow{(\hat{B}, \hat{A}, C_{\hat{A}}, \Phi_{\hat{A}}, s_{\hat{A}})}$	
	$\xleftarrow{(\hat{A}, \hat{B}, C_{\hat{B}}, \Phi_{\hat{B}}, s_{\hat{B}})}$	
$k'_{\hat{B}} \leftarrow HPS.Priv(sk_{\hat{A}}, C_{\hat{B}})$		$k'_{\hat{A}} \leftarrow HPS.Priv(sk_{\hat{B}}, C_{\hat{A}})$
$\Phi'_{\hat{B}} \leftarrow \mathsf{Ext}(k'_{\hat{B}}, s_{\hat{B}})$		$\Phi'_{\hat{A}} \leftarrow \mathsf{Ext}(k'_{\hat{A}}, s_{\hat{A}})$
$\Phi_{\hat{B}} =^{?} \Phi'_{\hat{B}}$		$\Phi_{\hat{A}} =^{?} \Phi'_{\hat{A}}$
$sid = (\hat{A}, \hat{B}, C_{\hat{A}}, C_{\hat{B}}, \mathcal{I})$		$\overline{sid} = (\hat{B}, \hat{A}, C_{\hat{B}}, C_{\hat{A}}, \mathcal{R})$
$K \leftarrow k_{\hat{A}} \oplus k'_{\hat{B}}$		$K \leftarrow k'_{\hat{A}} \oplus k_{\hat{B}}$
$seed \leftarrow \hat{A} \parallel \hat{B} \parallel pk_{\hat{A}} \parallel pk_{\hat{B}} \parallel C_{\hat{A}} \parallel C_{\hat{B}} \parallel k_{\hat{A}} \parallel k'_{\hat{B}}$		
$k_s \leftarrow \mathsf{PRF}(K, seed)$		$k_s \leftarrow \mathsf{PRF}(K, seed)$

Figure 1: A generic adaptively-secure AKE construction from HPS

4.2 Security Analysis

Apparently, we use a Hash Proof System (HPS) to generate an encapsulated key k as an idea in [23], then we used that key to derive the PRF key to obtain the session key. We used the extractor to prevent the key k leakage.

Theorem 1. *Assuming that HPS is an ϵ_{hps}-universal hash proof system, $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(\nu, \epsilon_{\mathsf{Ext}})$-Strong Extractor, $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_{\mathsf{PRF}}$ is a $(q, t, \epsilon_{\mathsf{PRF}})$-secure pseudo-random function family. Then the proposed protocol is eCK-secure in the sense of Deffinition 8.*

The proof of above theorem could be found in Appendix A.

5 Instantiation from the DDH Assumption

We organized this section as follows. In Section 5.1, we show how to construct a hash proof system from the Decisional Diffie-Hellman (DDH) assumption. We follow [7, 23] in the instantiation of our protocol from DDH assumption. In Section 5.2, we apply the construction in Section 4 to a building block and obtain an adaptively DDH-based secure AKE scheme, depicted in Figure 2. In this section, we will show a comparison of our scheme with some existing AKE-secure scheme.

5.1 A DDH-Based HPS

Let $\langle q, G, g \rangle \leftarrow \mathcal{G}(1^\lambda)$ and let $g_1, g_2 \leftarrow_{\$} G$. Let $\Gamma : G \to \mathbb{Z}_q$ be an efficient injective mapping.

For any $u = (u_1, ..., u_n) \in G^n, n \in \mathbb{N}$ let $\hat{\Gamma}(u) = (\Gamma(u_1), ..., \Gamma(u_n)) \in \mathbb{Z}_q^n$. Obviously, $\hat{\Gamma}$ is also an injection. We will set up the a parameter *param* of the hash proof system mentioned in Section 2.3 as follows.

- $group = \langle q, G, g_1, g_2, n \rangle, \mathcal{C} = G \times G, \mathcal{V} = \{(g_1^r, g_2^r) : r \in \mathbb{Z}_q\}$ with witness $\omega = \mathbb{Z}_q$.
- $\mathcal{K} = \mathbb{Z}_q^n, \mathcal{SK} = (\mathbb{Z}_q \times \mathbb{Z}_q)^n, \mathcal{PK} = G^n$.
- For all $sk = (x_{i,1}, x_{i,2})_{i \in [n]} \in \mathcal{SK}$ we define $pk = (pk_i)_{i \in [n]} = \mu(sk) = (g_1^{x_{i,1}} g_2^{x_{i,2}})_{i \in [n]}$.
- For all $C = (u_1, u_2) \in \mathcal{C}$ we define $\Lambda_{sk}(C) = \hat{\Gamma}((u_1^{x_{i,1}} u_2^{x_{i,2}})_{i \in [n]})$.
- HPS.Gen(1^λ): Will generate sk and pk as mentioned above.
- HPS.Pub$(pk, C, r) = \hat{\Gamma}(pk_1^r, ..., pk_n^r)$ for all $C = (g_1^r, g_2^r) \in \mathcal{V}$ with witness $r \in \mathbb{Z}_q$.
- HPS.Priv$(sk, C) = \Lambda_{sk}(C) = \hat{\Gamma}((g_1^{rx_{i,1}} u_2^{rx_{i,2}})_{i \in [n]})$ for all $C = (g_1^r, g_2^r) \in G^2$.
- Correctness follow since

$$\begin{aligned}
\hat{\Gamma}(pk_i^r)_{i \in [n]} &= \hat{\Gamma}(pk_1^r, ..., pk_n^r) = \hat{\Gamma}(g_1^{rx_{1,1}} g_2^{rx_{1,2}}, ..., \\
&\qquad g_1^{rx_{n,1}} g_2^{rx_{n,2}}) \\
&= \hat{\Gamma}(u_1^{x_{1,1}} u_2^{x_{1,2}}, ..., u_1^{x_{n,1}} u_2^{x_{n,2}}) \\
&= \Lambda_{sk}(C).
\end{aligned}$$

$\hat{A}$		$\hat{B}$
$(a_{i,1}, a_{i,2})_{i\in[n]} \leftarrow_\$ (\mathbb{Z}_q \times \mathbb{Z}_q)^n$		$(b_{i,1}, b_{i,2})_{i\in[n]} \leftarrow_\$ (\mathbb{Z}_q \times \mathbb{Z}_q)^n$
$A = (g_1^{a_{i,1}} g_2^{a_{i,2}})_{i\in[n]}$		$B = (g_1^{b_{i,1}} g_2^{b_{i,2}})_{i\in[n]}$
$x \leftarrow_\$ \mathbb{Z}_q, X = (X_1, X_2) \leftarrow (g_1^x, g_2^x)$		$y \leftarrow_\$ \mathbb{Z}_q, Y = (Y_1, Y_2) \leftarrow (g_1^y, g_2^y)$
$K_{\hat{A}} = \hat{\Gamma}(B_i^x)_{i\in[n]}$		$K_{\hat{B}} = \hat{\Gamma}(A_i^x)_{i\in[n]}$
$s_{\hat{A}} \leftarrow_\$ \{0,1\}^d$		$s_{\hat{B}} \leftarrow_\$ \{0,1\}^d$
$\Phi_{\hat{A}} \leftarrow \mathsf{Ext}(k_{\hat{A}}, s_{\hat{A}})$		$\Phi_{\hat{B}} \leftarrow \mathsf{Ext}(k_{\hat{B}}, s_{\hat{B}})$
$sid = (\hat{A}, \hat{B}, X, -, \mathcal{I})$		$\overline{sid} = (\hat{B}, \hat{A}, Y, -, \mathcal{R})$
	$\xrightarrow{(\hat{B}, \hat{A}, X, \Phi_{\hat{A}}, s_{\hat{A}})}$	
	$\xleftarrow{(\hat{A}, \hat{B}, Y, \Phi_{\hat{B}}, s_{\hat{B}})}$	
$k'_{\hat{B}} = \Lambda_{sk}(Y) = \hat{\Gamma}(Y_1^{a_{i,1}} Y_2^{a_{i,2}})_{i\in[n]}$		$k'_{\hat{A}} = \Lambda_{sk}(X) = \hat{\Gamma}(X_1^{b_{i,1}} X_2^{b_{i,2}})_{i\in[n]}$
$\Phi'_{\hat{B}} \leftarrow \mathsf{Ext}(k'_{\hat{B}}, s_{\hat{B}})$		$\Phi'_{\hat{A}} \leftarrow \mathsf{Ext}(k'_{\hat{A}}, s_{\hat{A}})$
$\Phi_{\hat{B}} =^? \Phi'_{\hat{B}}$		$\Phi_{\hat{A}} =^? \Phi'_{\hat{A}}$
$sid = (\hat{A}, \hat{B}, X, Y, \mathcal{I})$		$\overline{sid} = (\hat{B}, \hat{A}, Y, X, \mathcal{R})$
$K \leftarrow k_{\hat{A}} \oplus k'_{\hat{B}}$		$K \leftarrow k'_{\hat{A}} \oplus k_{\hat{B}}$
$seed \leftarrow \hat{A} \parallel \hat{B} \parallel pk_{\hat{A}} \parallel pk_{\hat{B}} \parallel C_{\hat{A}} \parallel C_{\hat{B}} \parallel k_{\hat{A}} \parallel k'_{\hat{B}}$		
$k_s \leftarrow \mathsf{PRF}(K, seed)$		$k_s \leftarrow \mathsf{PRF}(K, seed)$

Figure 2: A DDH-based adaptively-secure AKE construction from HPS

Theorem 2. *The above system* ***HPS****, which contains of following algorithms (HPS.Gen,HPS.Pub,HPS.Priv) is a ϵ-universal HPS for $\mathcal{V}$.*

The proof of above theorem could be found in Appendix A.

5.2 The DDH-Based Instantiation AKE Scheme from Scheme in Section 4

Parameters

Let $\mathbb{G} = \langle q, G, q\rangle$. Let $n \in \mathbb{N}$. Let HPS is ϵ-universal hash proof system described in above. Let $\mathsf{Ext} : \mathbb{Z}_q^n \times \{0,1\}^d \to \{0,1\}^m$ be a $(\nu, \epsilon_{\mathsf{Ext}})$-Strong Extractor. Let $\mathsf{PRF} : \mathbb{Z}_q^n \times \mathcal{D}_{\mathsf{PRF}} \to \mathcal{R}_{\mathsf{PRF}}$ be a $(q, t, \epsilon_{\mathsf{PRF}})$-secure pseudorandom function family. We obtain a DDH-based scheme as follows.

Key Generation

A party $\hat{A}$ picks $(a_{i,1}, a_{i,2})_{i\in[n]} \leftarrow_\$ (\mathbb{Z}_q \times \mathbb{Z}_q)^n$ and sets $A = (g_1^{a_{i,1}} g_2^{a_{i,2}})_{i\in[n]}$. The public/secret key for the party $\hat{A}$ is $((a_{i,1}, a_{i,2})_{i\in[n]}, A = (A_i)_{i\in[n]})$. Similarly, a party $\hat{B}$ will set his public/secret keys as $((b_{i,1}, b_{i,2})_{i\in[n]}, B = (B_i)_{i\in[n]})$. We assume this protocol executed between party $\hat{A}$ and party $\hat{B}$, where party $\hat{A}$ is the initiator and party $\hat{B}$ is the responder.

Messages Exchange

The party $\hat{A}$ chooses $x \leftarrow_\$ \mathbb{Z}_q$ and compute X as following $(X_1, X_2) \leftarrow (g_1^x, g_2^x)$, a random seed $s_{\hat{A}} \leftarrow_\$ \{0,1\}^d$ and then computes

$$K_{\hat{A}} = \hat{\Gamma}(B_i^x)_{i\in[n]}, \omega_{\hat{A}}), \Phi_{\hat{A}} \leftarrow \mathsf{Ext}(k_{\hat{A}}, s_{\hat{A}})$$

Then sends $(\hat{B}, \hat{A}, X, \Phi_{\hat{A}}, s_{\hat{A}})$ to party $\hat{B}$. Simulatly, the part $\hat{B}$ will follow the same steps and sends $(\hat{A}, \hat{B}, Y, \Phi_{\hat{B}}, s_{\hat{B}})$ to party $\hat{A}$.

Upon receiving $(\hat{B}, \hat{A}, X, \Phi_{\hat{A}}, s_{\hat{A}})$, party $\hat{B}$ uses his secret key to get $k'_{\hat{A}} = \Lambda_{sk}(X) = \hat{\Gamma}(X_1^{b_{i,1}} X_2^{b_{i,2}})_{i\in[n]}$, then computes $\Phi'_{\hat{A}} \leftarrow \mathsf{Ext}(k'_{\hat{A}}, s_{\hat{A}})$. The party $\hat{B}$ checks $\Phi_{\hat{A}} = \Phi'_{\hat{A}}$, if not then halt. Otherwise, the party $\hat{B}$ computes the session key.

Session Key

The party $\hat{B}$ compute his session key as following:

1) $K \leftarrow k'_{\hat{A}} \oplus k_{\hat{B}}$.

2) $seed \leftarrow \hat{A} \parallel \hat{B} \parallel pk_{\hat{A}} \parallel pk_{\hat{B}} \parallel C_{\hat{A}} \parallel C_{\hat{B}} \parallel k'_{\hat{A}} \parallel k_{\hat{B}}$.

3) $k_s \leftarrow \mathsf{PRF}(K, seed)$.

The correctness of the above protocol follows from the correctness of the underlying hash proof system.

Theorem 3. *If the DDH assumptions hold in groups G and $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(\nu, \epsilon_{\mathsf{Ext}})$-Strong*

Table 1: Protocols comparison

Protocol	**Computation**	**Model**	**Security**	**Assumption**	**NAXOS Approch**	**SPK/EPK**
Okamoto [22]	8E	eCK	Standard	DDH & πPRF	Yes	6/6
Moriyama *et al.* [21]	18E	eCK	Standard	DDH & πPRF	No	6/3
Fujioka *et al.* [9]	9E	eCK	Standard	DDH & KEM	No	1/2
HMQV [17]	2.5E	CK, wPFS, KCI, LEP	RO	GDH, KEA1	Yes	1/1
NAXOS [18]	4E	eCK	RO	GDH	Yes	1/1
Mojahed *et al.* [20]	3E	eCK	RO	DLIN	No	1/1
Our	4E	eCK	Standard	DDH,Λ_{sk}	No	2/1

Extractor, $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \rightarrow \mathcal{R}_{\mathsf{PRF}}$ *is a* $(q, t, \epsilon_{\mathsf{PRF}})$*-secure pseudo-random function family. Then the proposed protocol is eCK-secure in the sense of Deffinition 8.*

Proof. From the proof of Theorem 2 and according to proof of Theorem 1, we concludes the proof of Theorem 3. □

5.3 Efficiency

We show the efficiency of our protocol compare to other related ones regarding based assumption, computational efficiency and security model will be discussed in this section. In Table 1, we show number of exponentiation in G (E), number of static public keys (SPK) and number of ephemeral public key (EPK).

From Table 1, we show that our paradigm is much efficient group exponentiations count comparing to a similar protocol that does not rely on NAXOS trick or proved in the standard model. Our protocol does not rely on πPRF or KEM; it uses an adaptive smooth projective hash function instead. Since our protocol is using standard assumption and preliminaries, thus, it is practical to design it using different language programs and various devices.

6 Conclusions

In this paper, we presented a general construction for AKE-secure protocol from the projective hash family secures under hard subset membership problem in the standard model. We gave a novel security proof from games sequences tool introduced by [24]. Our methodology in research was how to design an eCK-secure paradigm from a smooth projective hash function defined in [7]. In our study, we gave a literature about using NAXOS trick in developing an AKE-secure protocol and stated open problem related to that. We proved the security of our paradigm in the standard model which presents another challenge in our research.

Moreover, we also gave an instantiation of our protocol from DDH. We show the efficiency of our protocol compares to other similar AKE protocol.

References

[1] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology (CRYPTO'93)*, pp. 232–249, Springer, 1993.

[2] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," in *IMA International Conference on Cryptography and Coding*, LNCS 1355, pp. 30–45, Springer, 1997.

[3] S. Blake-Wilson and A. Menezes, "Entity authentication and authenticated key transport protocols employing asymmetric techniques," in *Security Protocols*, pp. 137–158, Springer, 1997.

[4] E. Bresson, O. Chevassut, and D. Pointcheval, "Dynamic group diffie-hellman key exchange under standard assumptions," in *Advances in Cryptology (EUROCRYPT'02)*, pp. 321–336, Springer, 2002.

[5] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology (EUROCRYPT'01)*, pp. 453–474, Springer, 2001.

[6] O. Chevassut, P.-A. Fouque, P. Gaudry, and D. Pointcheval, "Key derivation and randomness extraction," *IACR Cryptology ePrint Archive*, vol. 2005, pp. 61, 2005.

[7] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *Advances in Cryptology (EUROCRYPT'02)*, pp. 45–64, Springer, 2002.

[8] C. J. Cremers, "Session-state reveal is stronger than ephemeral key reveal: Attacking the naxos authenticated key exchange protocol," in *Applied Cryptography and Network Security*, pp. 20–33, Springer, 2009.

[9] A. Fujioka, K. Suzuki, K. Xagawa, and K. Yoneyama, "Strongly secure authenticated key exchange from factoring, codes, and lattices," *Designs, Codes and Cryptography*, vol. 76, no. 3, pp. 469–504, 2015.

[10] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange," in *Advances in Cryptology (EUROCRYPT'03)*, pp. 524–543, Springer, 2003.

[11] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *Journal of the ACM*, vol. 33, no. 4, pp. 792–807, 1986.

[12] H. Huang, "Authenticated key exchange protocol under computational diffie–hellman assumption from trapdoor test technique," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 325–343, 2015.

[13] J. Katz and V. Vaikuntanathan, "Round-optimal password-based authenticated key exchange," *Journal of Cryptology*, vol. 26, no. 4, pp. 714–743, 2013.

[14] E. Kiltz, K. Pietrzak, M. Stam, and M. Yung, "A new randomness extraction paradigm for hybrid encryption," in *Advances in Cryptology (EUROCRYPT'09)*, pp. 590–609, Springer, 2009.

[15] M. Kim, A. Fujioka, and B. Ustaoğlu, "Strongly secure authenticated key exchange without naxosapproach," in *Advances in Information and Computer Security*, pp. 174–191, Springer, 2009.

[16] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology (CRYPTO'99)*, pp. 388–397, Springer, 1999.

[17] H. Krawczyk, "HMQV: A high-performance secure diffie-hellman protocol," in *Advances in Cryptology (CRYPTO'05)*, pp. 546–566, Springer, 2005.

[18] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security*, pp. 1–16, Springer, 2007.

[19] K. Minkyu, A. Fujioka, B. USTAO *et al.*, "Strongly secure authenticated key exchange without naxos'approach under computational diffie-hellman assumption," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 95, no. 1, pp. 29–39, 2012.

[20] M. Mohamed, X. Wang, and X. Zhang, "Efficient secure authenticated key exchange without naxos approach based on decision linear problem," in *Collaborative Computing: Networking, Applications, and Worksharing*, pp. 243–256, Springer, 2015.

[21] D. Moriyama and T. Okamoto, "An eCK-secure authenticated key exchange protocol without random oracles," in *International Conference on Provable Security*, LNCS 5848, pp 154–167, Springer, 2009.

[22] T. Okamoto, "Authenticated key exchange and key encapsulation in the standard model," in *Advances in Cryptology (ASIACRYPT'07)*, pp. 474–484, Springer, 2007.

[23] B. Qin and S. Liu, "Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter," in *Advances in Cryptology (ASIACRYPT'13)*, pp. 381–400, Springer, 2013.

[24] V. Shoup, "Sequences of games: A tool for taming complexity in security proofs." *IACR Cryptology ePrint Archive*, vol. 2004, pp. 332, 2004.

[25] V. Shoup, *On Formal Models for Secure Key Exchange*, Citeseer, 1999.

[26] Z. Yang, "Efficient eck-secure authenticated key exchange protocols in the standard model," in *Information and Communications Security*, pp. 185–193, Springer, 2013.

A Proof of Theorem 1

Let $\mathcal{M}$ be a polynomial bounded adversary against protocol Π.Let sid^* is the target session chosen by adversary $\mathcal{M}$. Without lose fo generality, assume $\hat{A}$ is the owner of the session sid^* and $\hat{B}$ is the peer. Let sid be $(\hat{A}, \hat{B}, \tilde{C}_{\hat{A}}, \tilde{C}_{\hat{A}}, I)$ where $\tilde{sk}_{\hat{A}} \leftarrow_{\$} \mathcal{SK}, \tilde{pk}_{\hat{A}} \leftarrow \mu(\tilde{sk}_{\hat{A}})$ is the public keys for $(\hat{A})$ and $\tilde{sk}_{\hat{B}} \leftarrow_{\$} \mathcal{SK}; \tilde{pk}_{\hat{B}} \leftarrow \mu(\tilde{sk}_{\hat{B}})$ is the public keys for $(\hat{B})$. Assume also that $\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(k)$ is the adversary advantage which we want to evaluate in this proof. From fresh session definition we will have this two events:

Case 1: Existence of a matching session $\overline{sid^*}$ for the target session sid^* which will give us following sub-events:

- Case 1.1 : $\neg StateReveal(sid^*) \vee \neg StateReveal(\overline{sid^*})$.
- Case 1.2 : $\neg StaticKeyReveal(owner) \vee \neg StaticKeyReveal(peer)$.
- Case 1.3 : $\neg StateReveal(sid^*) \vee \neg StaticKeyReveal(peer)$.
- Case 1.4 : $\neg StaticKeyReveal(owner) \vee \neg StateReveal(\overline{sid^*})$.

Case 2: No existence of a matching session for the target session sid^* which will generate the following sub-events:

- Case 2.1 : $\neg StateReveal(sid^*)$.
- Case 2.2 : $\neg StaticKeyReveal(owner)$.

Obviously, those sub-events are independent events. We can describe Case 1.2 as Case 2.2, similarly, we can describe cases (Case 1.3,Case 1.4) as Case 2.1. We do that because: the existance adversary can breaks the protocol in cases (1.2, Case 1.3, Case 1.4) will let us to construct another adversary can breaks it in cases (Case 2.2, Case 2.1). Thus, we can construct three adversaries will break the protocol in the previous sub-events.

Case 1.1

To analyze this event, Adversary $\mathcal{M}$ will play next games:

- Game_{1-0}: This is eCK original game where adversary $\mathcal{M}$ try to distinguish the real session key from random string.

Claim 1. *let* G0 *be the event that* $b = b'$ *in* Game_{1-0}. *we claim that*

$$\Pr[\mathsf{G0}] = \frac{\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(\lambda) + 1}{2} \qquad (2)$$

Proof. it's easy to derive the proof from Definition 8. □

- Game_{1-1}: This is reduced game from Game_{1-0}, In this game the adversary will choose only two parties $\hat{A},\hat{B}$ and only two sessions, the target session and its matching session$(sid^*, \overline{sid^*})$ with identifiers $(\hat{A},\hat{B},\tilde{C}_{\hat{A}},\tilde{C}_{\hat{B}},\mathcal{I})$ and $(\hat{B},\hat{A},\tilde{C}_{\hat{B}},\tilde{C}_{\hat{A}},\mathcal{I})$ respectively. We suppose that $\mathcal{M}$ activates at most $s(\lambda)$ sessions for each $n(\lambda)$ party For game state, see Appendix **??**.

Claim 2. *let* $\mathsf{G1}$ *be the event that* $\mathcal{A}$*success in guessing* sid^* *and* $\overline{sid^*}$ *in* Game_{1-1}*. we claim that*

$$\Pr[\mathsf{G0}] - \Pr[\mathsf{G1}] \leq \frac{2}{n(\lambda)^2 s(\lambda)} \qquad (3)$$

Proof. In this game it's obvious that this game is similar to game Game_{1-1} except it required the adversary to guess target session and its matching session correctly to win this game. To select correct parties $\hat{A}$ nad $\hat{B}$, adversary should choose between $n(\lambda)$ parties the couple$(\hat{A},\hat{B})$, Let $\Pr\left[\hat{A}\cap\hat{B}\right]$ denotes that event, thus:

$$\Pr\left[\hat{A}\cap\hat{B}\right] = \frac{1}{C_2^{n(\lambda)}} = \frac{1}{\frac{n(\lambda)!}{(n(\lambda)-2)!2!}}$$
$$= \frac{2}{n(\lambda)(n(\lambda)-1)} \leq \frac{2}{n^2(\lambda)}$$

In other hand, the adversary should success in guessing target session and its matching session.
Let $\Pr\left[sid^*_{\hat{A},\hat{B}} \cup sid^*_{\hat{A},\hat{B}}\right]$ denote the probability that adversary successfully guess the target session and its matching session thus:

$$\Pr\left[sid^*_{\hat{A},\hat{B}} \cup sid^*_{\hat{A},\hat{B}}\right] =$$
$$\Pr\left[sid^*_{\hat{A},\hat{B}}\right] + \Pr\left[sid^*_{\hat{A},\hat{B}}\right] - \Pr\left[sid^*_{\hat{A},\hat{B}} \cap sid^*_{\hat{A},\hat{B}}\right]$$

$$\Pr\left[sid^*_{\hat{A},\hat{B}} \cap sid^*_{\hat{A},\hat{B}}\right] =$$
$$\frac{1}{P_2^{s(\lambda)}} = \frac{1}{\frac{s(\lambda)!}{(s(\lambda)-2)!}}$$
$$= \frac{1}{s(\lambda)(s(\lambda)-1)}$$

thus

$$\Pr\left[sid^*_{\hat{A},\hat{B}} \cup sid^*_{\hat{A},\hat{B}}\right] =$$
$$\frac{1}{s(\lambda)} + \frac{1}{s(\lambda)} - \frac{1}{s(\lambda)(s(\lambda)-1)} =$$
$$\frac{s(\lambda)-2}{s(s(\lambda)-1)} \leq \frac{1}{s(\lambda)}$$

From these two probabilities, we can derive the whole probability that adversary success in guessing parties $\hat{A}$ and $\hat{A}$ with target session and its matching session with the form:

$$\Pr[\mathsf{G0}] - \Pr[\mathsf{G1}] \leq \Pr\left[\hat{A}\cap\hat{B}\right]\Pr\left[sid^*_{\hat{A},\hat{B}} \cup sid^*_{\hat{B},\hat{A}}\right]$$
$$= \frac{2}{n(\lambda)^2 s(\lambda)}$$

□

- Game_{1-2}: We transform Game_{1-1} into Game_{1-2}, the way of generation of k will be change.
In this game, Sim computes $k_{(.)}$ with HPS:Priv$(sk_{(.)}, C_{(.)})$ instead of HPS:Pub$(pk_{(.)}, C_{(.)}, \omega_{(.)})$.

Claim 3. *let* $\mathsf{G2}$ *be the event that* Sim *computes* $k_{(.)}$ *with HPS:Priv*$(sk_{(.)}, C_{(.)})$ *instead of HPS:Pub*$(pk_{(.)}, C_{(.)}, \omega_{(.)})$*. we claim that*

$$\Pr[\mathsf{G1}] = \Pr[\mathsf{G2}] \qquad (4)$$

Proof. Since HPS is projective, this change is purely conceptual, and thus Pr[S3] = Pr[S2]. □

- Game_{1-3}: We transform Game_{1-2} into Game_{1-3}, the way of generation of $\tilde{C}_{(.)}$ will be change. In this game, Sim samples $\tilde{C}_{(.)}$ from $\mathcal{C}\setminus\mathcal{V}$.

Claim 4. *let* $\mathsf{G3}$ *be the event that* $\tilde{C}_{(.)} \leftarrow_{\$} \mathcal{C}\setminus\mathcal{V}$*. we claim that*

$$\Pr[\mathsf{G2}] - \Pr[\mathsf{G3}] \leq \mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda) \qquad (5)$$

which $\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda)$ *is advantage of some efficient adversary* $\mathcal{A}$ *to beaks HPS.*

Proof. A straightforward reduction to the indistinguishability of the subset membership problem yields we can conclude our proof. □

- Game_{1-4}: We transform Game_{1-3} into Game_{1-4}. This game is the same as Game_{1-3}, except that we choose $\Phi_{(.)}$ randomly instead of computing it from the Ext function.

Claim 5. *let* $\mathsf{G4}$ *be the event of evaluation of* $\Phi_{(.)} \leftarrow_{\$} \{0,1\}^m$ *randomly. we claim that*

$$\Pr[\mathsf{G3}] - \Pr[\mathsf{G4}] \leq 2\epsilon_{\mathsf{Ext}} \qquad (6)$$

Proof. It shows clearly that if the adversary can distinguish between Game_{1-3} and Game_{1-4} then he can generate the same value of Ext function. In Game_{1-3}, $\Pr[G3]$ represents $\Pr\left[\mathsf{Ext}(k_{(.)}), s_{(.)}\right]$ where $s_{(.)}$ represent the seed of the extraction function and key generated from HPS. In Game_{1-4}, $\Pr[G4]$ represents $r \leftarrow_{\$} \{0,1\}^m$. From (k,$\epsilon$)-Strong Extractor definition we get

$$\frac{1}{2}\sum_{y\in(m,k)} \left|Pr[Ext(s,x) = \Phi_{(.)}] - Pr[r = \Phi_{(.)}]\right| = \epsilon$$

We can write the above equation in form

$$\frac{1}{2}\sum_{y\in(m,k)}|\Pr[G3]-\Pr[G4]|=\epsilon$$

which imply 6. □

- Game_{1-5}: We transform Game_{1-4} into Game_{1-5}. This game is the same as Game_{1-4}, except that we choose k_s randomly instead of computing it from the PRF function.

Claim 6. *let* G5 *be the event of computing* $k_s \leftarrow_{\$} \mathcal{R}_{\mathsf{PRF}}$ *randomly. we claim that*

$$\Pr[\mathsf{G4}]-\Pr[\mathsf{G5}]\leq\epsilon_{\mathsf{PRF}}+\frac{1}{2} \tag{7}$$

Proof. It shows clearly that if the adversary can distinguish between Game_{1-4} and Game_{1-5} then he can generate the same value of PRF function. $\Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\lambda)=1\right]$ represents $\Pr[\mathsf{G4}]-\Pr[\mathsf{G5}]$. From Pseudo-Random function definition we get

$$\left|\Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}}(\lambda)=1\right]-\frac{1}{2}\right|\leq\epsilon_{\mathsf{PRF}}$$

We can write above equation in form

$$|\Pr[\mathsf{G4}]-\Pr[\mathsf{G5}]|\leq\epsilon_{\mathsf{PRF}}+\frac{1}{2}$$

which complete the proof.
Apparently, Pseudo-Random function behave as one time pad in game Game_{1-5}, which imply

$$\Pr[\mathsf{G5}]=\frac{1}{2} \tag{8}$$

□

Combining Equations (3), (4), (5), (6), (7) and (8), we obtain:

$$\mathsf{Adv}^{\mathsf{ake}}_{\mathcal{M},\Pi}(\lambda)\leq\frac{4}{n^2(\lambda)s(\lambda)}+2\mathsf{Adv}^{\mathsf{smp}}_{\mathcal{A},HPS}(\lambda)+4\epsilon_{\mathsf{Ext}}+2\epsilon_{\mathsf{PRF}}+1 \tag{9}$$

From the sequence of preceding claims, and since those following probabilities $\mathsf{Adv}^{\mathsf{smp}}_{\mathcal{A},HPS}(\lambda), \epsilon_{\mathsf{Ext}}$ and ϵ_{PRF} are negligible in λ, then we conclude that $\mathsf{Adv}^{\mathsf{ake}}_{\mathcal{M},\Pi}(\lambda)$ is negligible in λ. Thus our protocol is secure.

Case 2

To analyze this event, Adversary $\mathcal{M}$will play next games:

- Game_{2-0}: This is eCK original game where adversary $\mathcal{M}$ try to distinguish the real session key from random string.

Claim 7. *let* G20 *be the event that* $b=b'$ *in* Game_{2-0}*. we claim that*

$$\Pr[\mathsf{G20}]=\frac{\mathsf{Adv}^{\mathsf{ake}}_{\mathcal{M},\Pi}(\lambda)+1}{2}. \tag{10}$$

Proof. That proof can be derived from Game_{1-0}. □

- Game_{2-1}: This is reduced game from Game_{2-0}, In this game the adversary will choose only two parties $\hat{A},\hat{B}$ and only target session (sid^*) with identifier $(\hat{A},\hat{A},\tilde{C}_{\hat{A}},\tilde{C}_{\hat{B}},\mathcal{I})$.

Claim 8. *let* G21 *be the event that* $\hat{A}$ *success in guessing* sid^* *in* Game_{2-1}*. we claim that*

$$\Pr[\mathsf{G20}]-\Pr[\mathsf{G21}]\leq\frac{2}{n^2(\lambda)s(\lambda)} \tag{11}$$

Proof. In this game, it's obvious that this game is similar to game Game_{2-1} except it's required adversary to guess target session correctly to win this game. To select correct parties $\mathcal{A}$nad $\mathcal{B}$, adversary should choose between $n(k)$ parties the couple$(\hat{A},\hat{B})$, Let $\Pr\left[\hat{A}\cap\hat{B})\right]$ denotes that event, thus:

$$\Pr\left[\hat{A}\cap\hat{B}\right]=\frac{1}{C_2^{n(\lambda)}}=\frac{1}{\frac{n(\lambda)!}{(n(\lambda)-2)!2!}}=\frac{2}{n(\lambda)(n(\lambda)-1)}\leq\frac{2}{n^2(\lambda)}$$

where C^a_b is the combination
In other hand, the adversary should success in guessing target session and its matching session. Let $\Pr\left[sid^*_{\hat{A},\hat{B}}\right]$ denote the probability that adversary successfully guess the target session from $s(\lambda)$ sessions, thus:

$$\Pr\left[sid^*_{\hat{A},\hat{B}}\right]=\frac{1}{s(\lambda)}$$

From these two probability we can derive the whole probability that adversary success in guessing parties $\hat{A}$ and $\hat{B}$ with target session and its matching session with the form:

$$\Pr[\mathsf{G20}]-\Pr[\mathsf{G21}]\leq\Pr\left[\hat{A}\cap\hat{B}\right]\Pr\left[sid^*_{\hat{A},\hat{B}}\cup sid^*_{\hat{B},\hat{A}}\right]=\frac{2}{n(\lambda)^2s(\lambda)}$$

□

- Game_{2-2}: This game is behave similiary to game Game_{1-2}. Thus, we concludes

$$\Pr[\mathsf{G1}]=\Pr[\mathsf{G2}] \tag{12}$$

- Game_{2-3}: We transform Game_{2-2} into Game_{2-3}, the way of generation of $\tilde{C}_{(.)}$ will be change. In this game, Sim samples $\tilde{C}_{(.)}$ from $\mathcal{C}\setminus\mathcal{V}$.

Claim 9. *let* G23 *be the event that* $\tilde{C}_{(.)} \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}$*. we claim that*

$$\Pr[\mathsf{G22}] - \Pr[\mathsf{G23}] \le \frac{q^2_{HPS.Priv}}{2}.\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda) \tag{13}$$

which $\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda)$ *is advantage of some efficient adversary* $\mathcal{A}$ *to beaks HPS, and* $q_{HPS.Priv}$ *is the number of queried made by* $\mathcal{A}$*on HPS.Priv.*

Proof. In this game we transformed from Game_{2-2} by changing $\tilde{C}_{(.)}$ with $\tilde{C}_{(.)} \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}$. Without losing of generality, The adversary will make $q_{HPS.Priv}$ queries to oracle without repeat of the same query. We can get the probability of halt as:

$$\begin{aligned}\Pr[\perp] = C_2^{q_{HPS.Priv}} &= \frac{q_{HPS.Priv}!}{(q_{HPS.Priv}-2)!2!}\\ &= \frac{q_{HPS.Priv}(q_{HPS.Priv}-1)}{2} \le \frac{q^2_{HPS.Priv}}{2}\end{aligned}$$

The difference between $\Pr[\mathsf{G22}]$ and $\Pr[\mathsf{G23}]$ can be parlayed into a corresponding $\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(\lambda)$. And that can be conclude clearly from the indistinguishability of the subset membership problem yields we can conclude. □

- Game_{2-4}: We transform Game_{2-3} into Game_{2-4}. This game is the same as Game_{2-3}, except that we choose $\Phi_{(.)}$ randomly instead of computing it from the Ext function.

Claim 10. *let* G24 *be the event of evaluation of* $\Phi_{(.)} \leftarrow_{\$} \{0,1\}^m$ *randomly,* ϵ_{Ext} *be the advantage that* $\mathcal{A}_{\mathsf{Ext}}$ *can breaks* Ext *security and* q_{Ext} *the number of queries made by* $\mathcal{A}_{\mathsf{Ext}}$*. We claim that*

$$\Pr[\mathsf{G23}] - \Pr[\mathsf{G24}] \le q^2_{\mathsf{Ext}}.\epsilon_{\mathsf{Ext}} \tag{14}$$

Proof. It shows clearly that if the adversary can distinguish between Game_{2-3} and Game_{2-4} then he can generate the same value of Ext function. In Game_{2-3}, $\Pr[G23]$ represents $\Pr\left[\mathsf{Ext}(k_{(.)}), s_{(.)}\right]$ where $s_{(.)}$ represent the seed of the extraction function and key generated from HPS. In Game_{2-4}, $\Pr[G24]$ represents $r \leftarrow_{\$} \{0,1\}^m$. From (k,$\epsilon$)-Strong Extractor definition we get

$$\frac{1}{2}\sum_{y\in(m,k)} \left|Pr[Ext(s,x) = \Phi_{(.)}] - Pr[r = \Phi_{(.)}]\right| = \epsilon$$

We can write the above equation in form

$$\frac{1}{2}\sum_{y\in(m,k)} |\Pr[G3] - \Pr[G4]| = \epsilon$$

Without losing of generality, We let the adversary to make q_{Ext} queries to oracle without repeat of the same query. We can get the probability of halt as:

$$\begin{aligned}\Pr[\perp] = C_2^{q_{\mathsf{Ext}}} &= \frac{q_{\mathsf{Ext}}!}{(q_{\mathsf{Ext}}-2)!2!}\\ &= \frac{q_{\mathsf{Ext}}(q_{\mathsf{Ext}}-1)}{2} \le \frac{q^2_{\mathsf{Ext}}}{2}\end{aligned}$$

combining above equations we conclude our proof. □

- Game_{2-5}: We transform Game_{2-4} into Game_{2-5}. This game is the same as Game_{2-4}, except that we choose k_s randomly instead of computing it from the PRF function.

Claim 11. *let* G25 *be the event of computing* $k_s \leftarrow_{\$} \mathcal{R}_{\mathsf{PRF}}$ *randomly,* ϵ_{PRF} *be the advantage that* $\mathcal{A}_{\mathsf{PRF}}$ *can breaks* PRF *security and* q_{PRF} *the number of queries made by* $\mathcal{A}_{\mathsf{PRF}}$*. We claim that*

$$\Pr[\mathsf{G24}] - \Pr[\mathsf{G25}] \le \frac{q^2_{\mathsf{PRF}}.\epsilon_{\mathsf{Ext}} + 1}{2} \tag{15}$$

Proof. It shows clearly that if the adversary can distinguish between Game_{2-4} and Game_{2-5} then he can generate the same value of PRF function. $\Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}_{\mathsf{PRF}}}(\lambda) = 1\right]$ represents $\Pr[\mathsf{G24}] - \Pr[\mathsf{G25}]$. From Pseudo-Random function definition we get

$$\left|\Pr\left[EXP^{ind-cma}_{\mathsf{PRF},\mathcal{A}_{\mathsf{PRF}}}(\lambda) = 1\right] - \frac{1}{2}\right| \le \epsilon_{\mathsf{PRF}}$$

We can write above equation in form

$$|\Pr[\mathsf{G4}] - \Pr[\mathsf{G5}]| \le \epsilon_{\mathsf{PRF}} + \frac{1}{2}$$

Without losing of generality, We let the adversary to make q_{PRF} queries to oracle without repeat of the same query. We can get the probability of halt as:

$$\begin{aligned}\Pr[\perp] = C_2^{q_{\mathsf{PRF}}} &= \frac{q_{\mathsf{PRF}}!}{(q_{\mathsf{PRF}}-2)!2!}\\ &= \frac{q_{\mathsf{PRF}}(q_{\mathsf{PRF}}-1)}{2} \le \frac{q^2_{\mathsf{PRF}}}{2}\end{aligned}$$

which complete the proof.
Apparently, Pseudo-Random function behave as one time pad in game Game_{1-5}, which imply

$$\Pr[\mathsf{G5}] = \frac{1}{2} \tag{16}$$

□

Combining 11,12,13,14,15 and 16 we obtain:

$$\begin{aligned}&\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(\lambda) \le\\ &\qquad \frac{4}{n^2(\lambda)s(\lambda)} + q^2_{HPS}\mathsf{Adv}^{\mathrm{smp}}_{\mathcal{A},HPS}(\lambda) +\\ &\qquad 2q^2_{\mathsf{Ext}}\epsilon_{\mathsf{Ext}} + q^2_{\mathsf{PRF}}\epsilon_{\mathsf{PRF}} + 1\end{aligned} \tag{17}$$

From the sequence of preceding claims, and since those following probabilities $\mathsf{Adv}^{\mathrm{smp}}_{\mathcal{A},HPS}(\lambda)$, ϵ_{Ext} and ϵ_{PRF} are negligible in λ, then we conclude that $\mathsf{Adv}^{\mathrm{ake}}_{\mathcal{M},\Pi}(\lambda)$ is negligible in λ. Thus our protocol is secure.

Appendix: Proof of Theorem 2

To prove this theorem we retrieve the definition of the universal projective hash function defined by Cramer and Shop [7]. To follow the previous definition of HPS, we should define following:

- Existence of a subset membership problem M.
- Existence of a ϵ-universal hash projective function.

Let M be a subset membership problem, we write $\Lambda[\mathcal{C},\mathcal{V},\mathcal{W},\mathcal{R}]$ to indicate the instance Λ where $\mathcal{C},\mathcal{V} = G^2, \mathcal{V} \subset \mathcal{C}, \mathcal{W} = \mathbb{Z}_q, \mathcal{R} = \mathbb{Z}_q^n$. For $(g_1^r, g_2^r) \in \mathcal{C}$ with witness $r \in \mathcal{W}$. We define two sequences of random variables as follows $C \leftarrow \mathcal{V}, C' \leftarrow \mathcal{C} \setminus \mathcal{V}$ at random.

Claim 12. *We say M is a hard subset membership problem if (Λ, C) and (Λ, C') are computationally indistinguishable.*

Proof. Let $C_0 = (u_1, u_2) = (g_1^r, g_2^r) \in \mathcal{V}$ where $r \in \mathbb{Z}_q$ is a valid witness. Let $C_1 = (u_1^*, u_2^*) \in \mathcal{C} \setminus \mathcal{V}$ where $u_1^*, u_2^* \leftarrow_{\$} G$. Retrieve the advantage of adversary in Section 2.3 we derive:

$$\begin{aligned}\mathsf{Adv}^{\mathrm{smp}}_{HPS,\mathcal{A}}(k) = |\Pr[\mathcal{A}(\mathcal{C},\mathcal{V},C_0) = 1 | C_0 \leftarrow_{\$} \mathcal{V}] \\ - \Pr[\mathcal{A}(\mathcal{C},\mathcal{V},C_1) = 1 | C_1 \leftarrow_{\$} \mathcal{C} \setminus \mathcal{V}]|\end{aligned}$$

Obviously, to distinguish between C_0 and C_1 is to solve the logarithm problem which is hard to solve by assumption. Thus, we say the subset membership is hard. □

Let $\Lambda_{sk} : \mathcal{C} \rightarrow \mathcal{K}$ be a projective hash function indexed with $sk \in \mathcal{SK}$ instantiated with $\Lambda_{sk}(C) = \hat{\Gamma}((u_1^{x_{i,1}} u_2^{x_{i,2}})_{i\in[n]})$.

Claim 13. *We say Λ_{sk} is an ϵ-universal projective hash function.*

Proof. To show the ϵ-universality we show that, for any fixed $C = (u_1, u_2) \in \mathcal{C} \setminus \mathcal{V}$ in the distripution of $(C, \Lambda_{sk}(C))$ is that of two random and independent group elements. Consider the map

$$f((x_{i,1}, x_{i,2})_{i\in[n]}) = (pk, C) = (g_1^{x_{i,1}} g_2^{x_{i,2}}, u_1^{x_{i,1}} u_2^{x_{i,2}})_{i\in[n]}$$

mapping trapdoor $sk = (x_{i,1}, x_{i,2})_{i\in[n]}$ to (pk, C) pairs. If we show that this map is injective then we are done, since we are applying this map to a random input and hence will get a random output. We will show an equivalent statement that the map $f'((x_{i,1}, x_{i,2})_{i\in[n]}) = \log_{g_1}(f((x_{i,1}, x_{i,2})_{i\in[n]})) = (\log g_1(pk), \log_{g_1}(C))$ is injective. Let

$$u_1 = g_1^{r_1}, u_2 = g_2^{r_2} = g_1^{\beta r_2}$$

for some $r_1 \neq r_2$ and $\beta = \log_{g_1}(g_2)$. We write

$$pk = (g_1^{x_{i,1}} g_2^{x_{i,2}})_{i\in[n]} = (g_1^{x_{i,1} + \beta x_{i,2}})_{i\in[n]}$$

$$C = (u_1^{x_{i,1}} u_2^{x_{i,2}})_{i\in[n]} = (g_1^{r_1 x_{i,1} + \beta r_2 x_{i,2}})_{i\in[n]}$$

so $(pk, C) = (g_1^{z_{i,1}}, g_1^{z_{i,2}})$ for

$$\begin{pmatrix} z_{1,1} \\ z_{2,1} \\ \vdots \\ z_{n,1} \\ z_{1,2} \\ \vdots \\ z_{n,2} \end{pmatrix} = \rho \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{n,1} \\ x_{n,2} \end{pmatrix}$$

where

$$\rho = \begin{pmatrix} 1 & \beta & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \beta & \cdots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & \beta \\ r_1 & r_2\beta & 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & r_1 & r_2\beta \end{pmatrix}$$

We can write it in the form $f'((x_{i,1}, x_{i,2})_{i\in[n]}) = \rho((x_{i,1}, x_{i,2})_{i\in[n]})^T$. Since $det(\rho) = \beta^n (r_2 - r_1)^n \neq 0$ the ρ is not singular, which shows that f' is an injective map and concludes the proof. □

Biography

Mojahed Ismail Mohamed currently, a PhD student of the School of Computer Science and Engineering at University of Electronic Science and Technology of China (UESTC), and have been with UESTC since Jan. 2011. He received his Master degree (2013) in Information Security from UESTC, Chengdu, China, and Bachelor degree (2005) in computer Engineering from Karary University. He did his researches in key exchange protocols (KE).

Xiaofen Wang currently, an associate professor of the School of Computer Science and Engineering at University of Electronic Science and Technology of China (UESTC), and have been with UESTC since Jan. 2010. She received her Ph.D. degree (2009) and Master degree (2006) in cryptography from Xidian University, Xi'an, China, under the supervision of Prof. Guozhen Xiao, and Bachelor degree (2003) in computer science from University of Electronic Science and Technology of China. She was a visiting research fellow at University of Wollongong, working with Prof. Yi Mu from Aug. 2014 to Aug. 2015. Her major research interests include Cryptography, Information Security, Network Security, and Data Security, etc.

Xiaosong Zhang He is the director of the big data research center and big data research institute of University of Electronic Science and Technology (UESTC). For his Bachelor, he graduated from Shanghai Jiaotong University, Master, Ph.D. from the University of Electronic

Science and Technology of China. He is a long-term commitment to network and information security, computer application technology research, focusing on cyberspace security, IT infrastructure, big data security and applications, embedded platform security, network attack detection and software vulnerability and other key areas and Support technology To carry out innovative research and technological research.

Traffic Characteristic Map-based Intrusion Detection Model for Industrial Internet

Rui-Hong Dong, Dong-Fang Wu, Qiu-Yu Zhang and Tao Zhang
(Corresponding author: Qiu-Yu Zhang)

School of Computer and Communication, Lanzhou University of Technology
No. 287, Lan-Gong-Ping Road, Lanzhou 730050, China
(Email: zhangqylz@163.com)

(Received June 4, 2017; revised Aug. 16, 2017 and accepted Sep. 25, 2017)

Abstract

After the Stuxnet security event in Iran, the security issues on industrial Internet are very serious. Besides, there are many flaws existing in the modern traffic modelling approaches to the industrial field network. Aiming at these problems, the traffic characteristic map-based intrusion detection model for industrial Internet was proposed. Firstly, information entropy method was adopted to select vital traffic characteristics attributes set which is used to form traffic characteristic vectors. Secondly, multiple correlation analysis approach was applied to transform traffic characteristics vector into triangle area mapping matrix and traffic characteristic map can be established. Finally, using discrete cosine transform (DCT) and singular value decomposition (SVD) methods, perceptual hash digest database of normal and abnormal traffic characteristics maps was obtained. Thereafter, the corresponding intrusion detection rule set can be generated, which is essential for the modelling of network traffic periodic characteristics in industrial field network. In particular, the robustness and discrimination of the traffic characteristics map perceptual hash algorithm (TCM-PH) were proved. Experimental results show that the proposed approach has a good performance of intrusion detection in the industrial field network.

Keywords: Hash Digest; Industrial Control Network; Intrusion Detection; Network Traffic Characteristics Map; Rule Set

1 Introduction

The traditional industrial control system (ICS) is widely used in many national critical infrastructures (NCI), for instance, petrochemical industry, power and water conservancy industry, industrial production, nuclear energy and transportation. According to the analysis of Security Situation Report of ICS-CERT [6] in 2015, more than 80% NCI rely on ICS to achieve the automation of production process. Therefore, ICS plays a vital role in our daily life. The ICS security issues directly affect national security and economic development. In 2010, the Stuxnet virus infected the Bushehr nuclear power station in Iran. Until 2015, a series of network security issues appeared, which brought great influence on human's life. The security situation of industrial Internet is very serious [1, 3, 12].

Industrial Internet includes three layers: enterprise management network, supervisory network and field network. In the research of security problems in field network, the periodicity characteristic of network traffic information is the key point. The intrusion detection methods to field network can be divided into three kinds: intrusion detection approaches based on model, fuzzing detection and Snort rule. The Snort-based method is mostly used to analyse the protocol used in the field network. However, this kind of method highly relies on the prior knowledge, which is mostly used in known attacks [18, 23, 26]. The fuzzing-based method is mainly used to test the protocol vulnerability [21].

Aiming at the above questions, from the image point, the intrusion detection issues and relationships between every two attributes are researched. The traffic characteristic map-based intrusion detection model for industrial Internet was proposed in this paper. The presented method can meet the real-time and high efficiency of intrusion detection approach. Traditional text information can be transformed into image via the traffic characteristics map technique. In order to research traffic characteristics from different point, the single attribute research is replaced by the research about the relations between attributes. By using image perceptual hash features extraction method, hash digest can be obtained and intrusion detection rule set can be produced. The perceptual hash features extraction method obtains robustness and discrimination. The robustness ensures that the intrusion detection approach can effectively find the known traffic information. The discrimination keeps the distinguishing characteristics of unknown attacks. Furthermore, the time complexity of image perceptual hash algorithm is very low. Finally, the training and test processes can take the testbed data set [9] and NSL-KDD [20].

This research deals with four issues. 1) By using traffic characteristics map technique, the traffic text information can be transformed into traffic characteristics map. 2) By using image perceptual hash features extraction method, the hash digest can be captured and intrusion detection rule set can be produced. 3) Intrusion detection rule matching operation includes three steps: strings-based precise match, similarity measure based on Hamming distance and clustering based on Euclidean distance. These three steps ensure the intrusion detection performance. In the intrusion detection stage, three-level detection pattern is set. The adaptability of the proposed method is increased. The unknown attacks can be detected, which decreases the false alarm rate resulted from fuzzy matching. Our method has a good detection performance. 4) The study proves the robustness and discrimination of TCM-PH algorithm.

The rest of the paper is organized as follows: In Section 2, the related works of intrusion detection method based on traffic characteristics were described. The theory of traffic characteristics map technique and the image perceptual hash features extraction method were introduced in Section 3. In Section 4, an industrial Internet intrusion detection model based on traffic characteristics map was proposed. And the robustness and discrimination of this algorithm are also proved. In Section 5, the experimental results were analysed then the performances of our method and other methods were compared. Finally, we conclude our paper in Section 6.

2 Related Works

In industrial field network, intrusion detection method based on model has adaptability, which is essential for the detection of unknown attacks. Therefore, many researchers achieve more works in intrusion detection method based on traffic periodicity characteristics. In [5], the research work of Modbus traffic periodicity features was finished. According to the deep analysis of packets, the deterministic finite automation (DFA) approach was used to establish Modbus normal behaviour model, which displays a good abnormal detection performance and adaptability. Yet, the study did not analyse the algorithm complexity of the DFA. In [22], the researchers analysed the normal traffic characteristics to get the Snort rule set. Then, traffic white list was set and the abnormal traffic information can be detected. But, the performance verification of the proposed approach should have taken a universal data set. In [23], according to the prior knowledge, fuzzing detection method was applied to analyse the structure of packets. Thereafter, the vulnerability can be found.

In [26], fuzzing method was used to produce large number of malformed packets including function code, which can be adopted to test the vulnerability of SCADA system. The data space was compressed and fuzzing test time was optimized. Mostly, the fuzzing test method was used to find the vulnerability of industrial field network protocol. In [21], Modbus traffic information and terminal unit information were extracted to produce Snort rules. The intrusion detection system based on Modbus protocol and Snort rule was established. But, the production of Snort rule highly relies on prior knowledge. In [28], network data was mapped into different dimension of hash histogram to establish detection vector. Support vector data description machine (SVDD) was used to detect network abnormal information. And the comparing works between several different classifiers were finished. In [16], by using PSO-SVM method, in Modbus protocol packets, the research of function code appearance frequency was achieved.

In [27], aiming at network traffic top-down time series characteristics, a network traffic analysis system based on multi-view was established. In [19], the multiple correlation analysis (MCA) method was used to transform text information into corresponding traffic images. The differences between two traffic images were computed by Manhattan distance, which realized the abnormal intrusion detection. By using MCA, the traffic data characteristics can be kept. For the abnormal detection in field network, in [24], cumulative sum (CUSUM) method was used to deeply analyse network packet. According to [15], multiscale principal component analysis (MSPCA) method was used to research traffic periodicity and traffic matrix space-time correlation, which modelled the network normal traffic behaviours.

In [11], the active degree of input and output traffic in researched network was counted and the active entropy was computed via the active entropy method. This method produced intrusion detection rule set, which reduced the false positive rate. By [7], the improved affinity propagation (AP) method decreased the number of clustering classes and the time cost and ensured true positive rate, which increased intrusion detection performance. In [17], the simulations of network intrusion detection attacks were added into the testbed data proposed by [9]. The existing intrusion detection methods were evaluated. In [10], the rule set was captured from the Modbus protocol contents and traffic packet periodicity characteristics. And, the intrusion detection system was established. In [4], according to the research of registers value change characteristics in SCADA industrial control system, parameter models were established. These models can detect network abnormal traffic information.

Image perceptual hash features extraction approaches include DCT, SVD, wavelet transform and principal component analysis (PCA). In this paper, image perceptual hash features extraction methods based on SVD and DCT were adopted. In [8], colourful histogram and DCT coefficient matrix were regarded as perceptual features. The image contents tamper localization was achieved via DCT and PCA. This method demonstrated robustness and discrimination. In [2], the robustness characteristics were captured via audio cochleogram. And the established non-negative matrix was factorized to produce perceptual hash digest. By recurrence quantification method, the hash digest was matched. In [14], audio clips were mapped into hash digests. And the indexing and authentication of audios were achieved by this method. This method obtained robustness and discrimination.

3 Related Techniques and Theory

3.1 Modbus Traffic Characteristic Map Techniques

The approaches of information collection and features extraction in industrial Internet were researched. Based on above research works, the attributes features set of experimental data can be captured. The traffic information features space of Modbus protocol field network can be established. By using traffic characteristic map (TCM) technique, traffic characteristic map of field network traffic information can be produced. The traffic characteristic map is the input data for the image perceptual hash features extraction method.

Traffic information of field network has a strong periodicity, which leads to a fixed pattern. Traffic characteristics are much different between normal and abnormal traffics. The statistical characteristics of traffic information can be used to describe traffic behaviours. In [9], for the standardization of experimental data, a SCADA testbed experimental data was proposed. However, the testbed data and NSL-KDD [20] cannot be transformed into traffic characteristics map directly. Before transformation, the pre-processing works are needed to be finished. As the Figure 1 shown, the technique road of Modbus traffic characteristics map method is illustrated.

3.1.1 Compute Attributes Information Entropy and Normalization

In the pre-processing stage of experimental data, the incomplete traffic records are deleted. Then, the information entropy [25] of traffic attributes is computed. The vital attributes are selected. The definition of information entropy is as follow.

$$H(x) = -\sum_{i=1}^{s}(\frac{d_i}{T})log(\frac{d_i}{T}) \qquad (1)$$

where x is an attribute, and $H(x)$ is the information entropy of attribute x. The total number of attributes is n. The number of different traffic records is s. These traffic records can be expressed as $\{a_1, a_2, \cdots, a_s\}$. The corresponding occurrence number is $\{d_1, d_2, \cdots, d_s\}$. The computed information entropy of attributes can be sorted by descending order. The experimental data [9] includes the following data sets. Data 1 and Data 2 are collected from gas pipeline system. Data 3 and Data 4 are collected from water storage system. Data 1 and Data 3 are training data, and Data 2 and Data 4 are tested data. The traffic attributes number set of Data 1 and Data 2 is {1, 2, 3, 4, 5, 6, 12, 13, 24, 25, 26, 27}. The traffic attributes number set of Data 3 and Data 4 is {1, 2, 3, 4, 5, 6, 10, 12, 13, 18, 20, 21, 22, 23, 24}. The traffic attributes number set of NSL-KDD is {1,2,3,4,5,6,10,12,13,18,20,21,22,23,24}, the attributes tables are shown in Tables 3, 4 and 5.

The normalization of attributes set is defined as follow.

$$f(x) = \begin{cases} 0 & x \in [0, m) \\ \frac{255x}{n-m} & x \in [m, n] \\ 255 & x \in (n, \infty) \end{cases} . \qquad (2)$$

where n and m represent maximum and minimum, respectively. $f(x)$ is the normalization value, which is within the range of grey value, $f(x) \in [0, 255]$.

3.1.2 Multiple Correlation Analysis

In [19], adopting MCA method and triangle area method, normal and abnormal traffic characteristics were obtained. The correlations between attributes were also obtained. The flow steps are listed as follows.

Experimental data can be expressed as $X = \{x_1, x_2, \cdots, x_n\}$. According to the obtained traffic attributes sets, $x_i = [x_1, x_2, \cdots, x_n]$, $(1 \leq i \leq n)$, expresses the i-th m dimension traffic record. Triangle area method is used to capture the correlation between attributes j and k in vector x_j.

Vector x_j is mapped into $(j-k)$ dimension Euclidean subspace. $(1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq m, j \neq k)$, $y_{i,j,k} = [\varepsilon_j \varepsilon_k]^T = [f_j^i f_k^i]^T$, $\varepsilon_j = [e_{j,1}, e_{j,2}, .., e_{j,n}]$, $\varepsilon_k = [e_{k,1} e_{k,2} .. e_{k,n}]^T$, $e_{j,j} = e_{k,k} = 1$, and other elements equal to zero. $y_{i,j,k}$ is a 2-dimension vector, which can be expressed as one point $(f_j^i f_k^i)$ in $(j-k)$ dimension Euclidean subspace. On the Cartesian coordinate system, a triangle area $\Delta f_j^i O f_k^i$ is formed by the origin and the projected points of the coordinate $(f_j^i f_k^i)$ are found on the k and j axis. The triangle area can be expressed as following.

$$Tr_{j,k}^i = (\| (f_j^i, 0) - (0, 0) \| \times \| (0, f_k^i) - (0, 0) \|)/2. \qquad (3)$$

where $1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq m$, and $j \neq k$.

For the complete analysis of traffic records, x_i represents correlation between every two attributes. And, the corresponding triangle area is computed. The complete triangle area map (TAM) of traffic record including all triangle area is computed on the basis of every two attributes correlation. In i-th traffic record, $Tr_{j,k}^i$ expresses j-th row and k-th column triangle area. When $j = k$, $Tr_{j,k}^i = 0$. Therefore, the research focus on the correlations between every two attributes. When $j \neq k$, $Tr_{j,k}^i = Tr_{k,j}^i$. The obtained $\boldsymbol{TAM}$ is a symmetric matrix, whose main diagonal vector equal to zero. 4-dimension $\boldsymbol{TAM}$ can be expressed as follow.

$$TAM_x^i = \begin{bmatrix} 0 & Tr_{1,2}^i & Tr_{1,3}^i & Tr_{1,4}^i \\ Tr_{2,1}^i & 0 & Tr_{2,3}^i & Tr_{2,4}^i \\ Tr_{3,1}^i & Tr_{3,2}^i & 0 & Tr_{3,4}^i \\ Tr_{4,1}^i & Tr_{4,2}^i & Tr_{4,3}^i & 0 \end{bmatrix} \qquad (4)$$

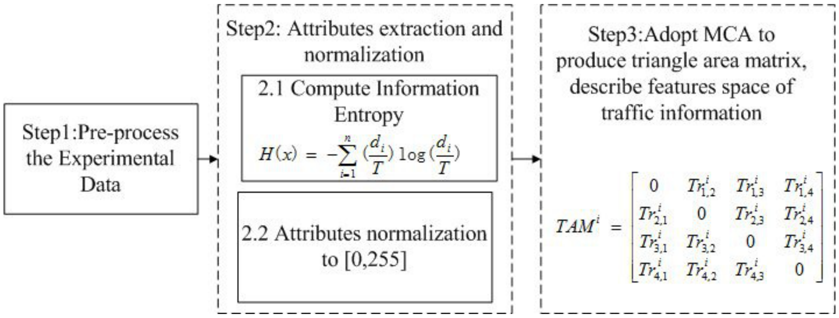

Step1:Pre-process the Experimental Data
Step2: Attributes extraction and normalization
2.1 Compute Information Entropy
$H(x) = -\sum_{i=1}^{n} (\frac{d_i}{T}) \log (\frac{d_i}{T})$
2.2 Attributes normalization to [0,255]
Step3:Adopt MCA to produce triangle area matrix, describe features space of traffic information
$TAM^i = \begin{bmatrix} 0 & Tr^i_{1,2} & Tr^i_{1,3} & Tr^i_{1,4} \\ Tr^i_{2,1} & 0 & Tr^i_{2,3} & Tr^i_{2,4} \\ Tr^i_{3,1} & Tr^i_{3,2} & 0 & Tr^i_{3,4} \\ Tr^i_{4,1} & Tr^i_{4,2} & Tr^i_{4,3} & 0 \end{bmatrix}$

Figure 1: Technique rode of Modbus traffic characteristics map method

There are several merits of MCA method. 1) There is no need of prior knowledge. 2) The MCA which is based on triangle area is not vulnerable to linear changes of all features. 3) Provide individual network traffic records. 4) Analyse the correlation between every two attributes.

Algorithm 1 TCM Algorithm

```
Input: Training data
Output: Traffic Characteristics Map $TAM^i$
According to Equation (1), compute the information entropy of attributes and select vital attributes set
According to Equation (2), normalize attributes set to $[0, 255]$
Get width and height of the dimension
for $i$ from 1 to height do
    for $j$ from 1 to width do
        According to Equation (3), compute $Tr^i_{j,k}$, and send value to the corresponding place in $\boldsymbol{TAM^i}$
    end for
end for
Output the traffic characteristics map $\boldsymbol{TAM^i}$
```

3.2 Image Perceptual Hash Features Extraction

3.2.1 Discrete Cosine Transform

DCT method has several merits: explicit physical meaning, middle complexity, swift calculation and separable property. DCT is regarded as the optimization method used in audio and image transformation. The transformation of image is achieved via DCT. According to [8], DCT can be defined as follow.

$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{x=0}^{N} \sum_{y=0}^{N} f(x,y) \cdot \cos(\frac{\pi(2x+1)u}{N^2}) \cos(\frac{\pi(2y+1)v}{N^2}) \quad (5)$$

where f expresses $N \times N$ pixels matrix, and F is $N \times N$ coefficient matrix. C expresses the cosine coefficient matrix.

The steps of algorithm as follow:

1) In Data 1, Data 2 and NSL-KDD train data set, by using TCM method, the 11×11, 14×14 and 14×14 traffic characteristics maps are obtained. And, in Data 3, Data 4 and NSL-KDD test data set, the 11×11, 14×14 and 14×14 traffic characteristics maps are got.

2) According to Equation (5), by using DCT method, 11×11, 14×14 and 14×14 DCT coefficient matrices are obtained.

3) For the discrimination of perceptual hash digest, the complete DCT coefficient matrix including low and high frequency domains is used to produce hash digest. And, the mean value of coefficient matrix is computed, named ***mean***.

4) According to Equation (7), SVD is used to decompose and reconstruct DCT coefficient matrix. The useful information can be obtained and the data noise is removed. $N = 11$ and 14, the left singular value u_3 and the right singular v_3, which are corresponded to $_3$, are used to produce hash digest ***DCT_m***.

$$DCT\_m = [u_1, u_2, .., u_N] \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix} [v_1, v_2, .., v_N]^T \quad (6)$$

$$SVD\_matrix = [u_3^T, v_3^T] \quad (7)$$

5) From left to right, the traversal work of SVD_matrix is achieved. The hash rule is defined as follow.

$$h(x) = \begin{cases} 1, & x \geq m \\ 0, & x < m \end{cases}. \quad (8)$$

where x is the SVD result, and $h(x)$ is the corresponding hash code. After the matrix traversal, the hash digests database can be obtained. And, the corresponding rule set is produced.

3.2.2 Match Hash Digest

The hash digests of test data are captured. These hash digests are the input data for hash match algorithm. By this way, the abnormal traffic information can be detected.

The image contents match methods include Euclidean distance, Hamming distance and norm. Hamming distance method is adapted to measure similarity of different images. Before hash matching, by using traffic characteristics map method, traffic text information can be transformed into traffic characteristics map. There are three matching stages in the improved matching algorithm. The three-level matching method increases adaptability of the TCM-PH algorithm.

1) Precise matching based on string

   The hash digest of test data is captured, named H_{s1}. Adopting the precise matching method, the normal traffic hash digest H_{s2} and abnormal traffic hash digest H_{s3} are matched with H_{s1}. Then, the matching results are output. After the precise matching, some traffic hash digests don't have any detection result. These hash digests take part in the second matching stage.

2) Similarity matching based Hamming distance

   By using Hamming distance Equation (9), the similarity between H_{s1} and normal traffic H_{s2} can be computed. And, the distance between H_{s1} and H_{s3} can also be calculated. According to the computed results, in condition to meet the match threshold value, the H_{s1} can obtain the matching result. However, some unknown intrusion traffic digests cannot meet this threshold. These unknown attacks participate in the third matching stage. The similarity can be expressed by D_H or bit error rate BER.

   $$D_H(H_{s1}, H_{s2}) = \frac{\sum_{w=1}^{L} \mid H_{s1}(w) - H_{s2}(w) \mid}{L} \tag{9}$$

   where $BER = D_H(H_{s1}, H_{s2})$, H_{s1} and H_{s2} have equal length. In Data 1 and Data 2, $L = 22$. In Data 3 and Data 4, $L = 28$, In NSL-KDD, $L = 28$. w is one hash code in hash digest. The threshold value of similarity matching is set.

3) Clustering matching based on Euclidean distance

   The Euclidean distance can be expressed as follow.

   $$d(x, y) = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^2} \tag{10}$$

   According to Equation (10), the distance from hash digest clustering centre to test hash digests can be computed. The test digest can be detected by the smallest Euclidean distance. The clustering method has adaptability, which is essential for the detection of unknown intrusion.

4 The Proposed Model

4.1 Intrusion Detection Model Based on TCM-PH

In the existing intrusion detection methods to industrial Internet, the nature of intrusion detection method based on traffic is to find the abnormal change rules of traffic. By the establishment of network traffic characteristics map in SCADA and field network, the intrusion detection model based on traffic characteristics map is established. The network intrusion detection issues are solved via image features extraction methods. Traffic characteristics map perceptual hash (TCM-PH) algorithm is a supervisory learning method. The intrusion detection thought is illustrated in Figure 2.

Algorithm 2 TCM-PH

```
Input: Training data and test data
Output: Intrusion detection results
Get training data set.
while the number of training data do
    The TCM algorithm is adopted to produce traffic
    characteristics map.
    By DCT method, the normal and abnormal hash
    digest are captured.
    Produce intrusion detection rule set.
end while
Get test traffic data set.
while test data do
    By TCM algorithm, the traffic characteristics map
    is produced.
    Adopting DCT method, the hash digest is captured.
    if meet precise matching then
        while rule do
            Hash digest match rule set, output detection
            result
        end while
    else if Meet similarity measure then
        According to Equation (9), the similarity between
        hash digest and intrusion detection rule
        set is computed.
        if matching threshold then
            Output intrusion detection result
        else if Meet similarity measure then
            According to Equation (10), compute distance
            between hash digest and clustering centre, output
            intrusion detection result
        end if
    end if
end while
```

4.2 The Property Proof of TCM-PH

When perceptual hash is applied in network intrusion detection, the robustness and discrimination [8] of TCM-PH

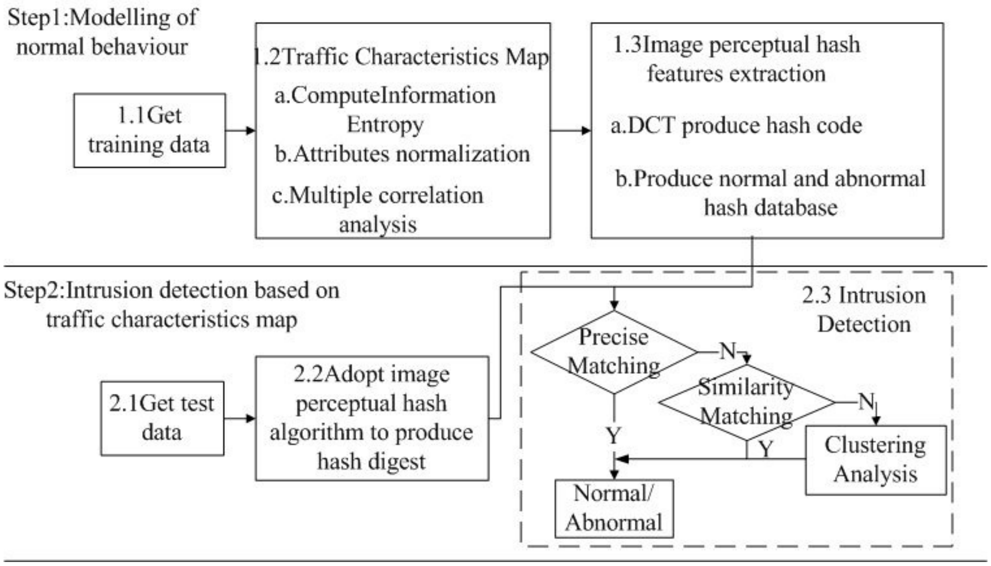

Step1:Modelling of normal behaviour
1.1Get training data
1.2Traffic Characteristics Map
a.ComputeInformation Entropy
b.Attributes normalization
c.Multiple correlation analysis
1.3Image perceptual hash features extraction
a.DCT produce hash code
b.Produce normal and abnormal hash database
Step2:Intrusion detection based on traffic characteristics map
2.1Get test data
2.2Adopt image perceptual hash algorithm to produce hash digest
2.3 Intrusion Detection
Precise Matching
N
Similarity Matching
N
Y
Y
Clustering Analysis
Normal/ Abnormal

Figure 2: Intrusion detection model based on traffic characteristics map

algorithm are proved. The perceptual hash function has robustness and discrimination [13]. The details are illustrated as follows.

Property 1. *(**Robustness**) After the content hold operations, the different media digital representation which possessed same perceptual content can produce one same hash digest.*

Property 2. *(**Discrimination**) The different media digital representation which possessed different content produce different hash digests.*

Assume that $x, y, z \in M$ is experimental data, and h_x, h_y, $h_z \in H_p$ are normal and abnormal traffic hash digests. $h_x = PH(x), h_y = PH(y), h_z = PH(z)$, PH is hash function. $dis(\cdot,\cdot)$ is perceptual distance (or false accept rate), and τ is matching threshold value. $dis(\cdot,\cdot)$ is geometric distance, and T_p is perceptual threshold. In data $\boldsymbol{M}$, x is the traffic characteristics map transformed from traffic text information. In 3.2 Section, traffic characteristics map method is illustrated carefully. x can be expressed as Equation (4). By DCT perceptual hash function, the hash digest is produced, and the hash digest can be expressed as $h_x = \{h_{x1}, h_{x2}, h_{x3}, hx4\}$.

The theory of geometric distance is Hamming distance in Equation (9). The matching threshold τ meet the range $(0,1]$. The perceptual distance describes the differences between multi-media data, which can be defined as follows.

$$disp(x,y) = \begin{cases} 1, & x \neq y \\ 0, & x = y \end{cases}. \tag{11}$$

where perceptual distance $T_p \in (0,1)$. When x and y are same $x = y$, $disp(x,y) < T_p$. When x and y are different, $x \neq y$, $disp(x,y) > T_p$. Let assume that $x = y$ and $x \neq z$.

Prove 1. *(**Robustness**) When the perceptual hash-based intrusion detection method has robustness, $\forall x, y \in M$, event $A = \{(x,y) : disp(x,y) < T_p$ and $dis(h_x,h_y) < \tau)\}$, $P(A) = 1$.*

It demonstrates the fact that x and y are same, $x = y$. Therefore, $disp(x,y) = 0$, $disp(x,y) < T_p$. When the perceptual hash is used in intrusion detection, there is no any content keeping manipulations. According to the robustness of perceptual hash function, the same media digital representation map into the same hash digest, $h_x = h_y$. According to Equation (10), $dis(h_x,h_y) = 0$, $dis(h_x,h_y) < \tau$. So, the probability of event A is 1, $P(A) = 1$. It means that x and y produce the same hash digest. The robustness of TCM-PH is proved.

Prove 2. *(**Discrimination**) When the perceptual hash-based intrusion detection method has discrimination, $\forall x, z \in M$, $B = \{(x,z) : disp(x,z) > T_p$ and $dis(h_x,h_z) < \tau\}$, $P(B) = 0$.*

Let assume that $P(B) = 1$, $dis(h_x,h_z) < \tau$, according to Equation (10), $h_x = h_z$. Considering the robustness of TCM-PH algorithm, we can judge that x and z are same traffic data, $x = z$. And, according to $P(B) = 1$, we can learn that $disp(x,z) > T_p$, by perceptual distance Equation (11). Therefore, x and z are different traffic data. According to the theory of reduction to absurdity, there is a contradiction in the mathematical reasoning. In fact, the original mathematical hypothesis is wrong. And, $P(B) = 0$. The discrimination of TCM-PH algorithm is proved, which ensure that different traffic data has different hash digest.

5 Experimental Results and Analysis

5.1 Selection of Experimental Data

The experimental data [9] and NSL-KDD [20] are adopted in our research to test the performance of TCM-PH algorithm. There are three merits in the testbed data. 1) This data provide more research opportunities with general researchers. 2) The data ensure that other researchers can test more vital studies and experimental results. 3) The proposed data provide the general test platform, which is helpful for researchers to compare and analyse every related methods. The merits of NSL-KDD data set include: 1) The redundancy data and the repeated data were removed for the objective evaluation. 2) The percentage and kinds of the records are same with the KDD Cup 99 data set. 3) The number of the NSL-KDD is feasible which decreases the payloads of the intrusion detection method.

Comparing with KDD99 data set, the proposed data [9] also sign every record with information type in 0 to 7 numbers. There are 8 kinds of traffic records. 0 represents normal data. Other numbers represent attack information. The normal traffic information is captured from the testbed SCADA system. And, abnormal traffic information can be divided into four kinds, reconnaissance, response injection, command injection and denial-of-service. There are total eight kinds of testbed data in Table 1.

NSL-KDD data is the improved version of the original KDD99 data set. Each value of label expresses different kinds of data. 0 is normal and others are abnormal. NSL-KDD includes four kinds of abnormal: Dos, Probe, U2R and R2L, shown in Table 2.

Table 2: The kinds of NSL-KDD data set

Label	Value	Description
Normal	0	Normal data
Dos	1	Deny of service attack
Probe	2	Probe attack
U2R	3	User to root attack
R2L	4	Remote to login attack

The features selection result of gas data set is $\{1, 2, 3, 4, 5, 6, 12, 13, 24, 25, 26, 27\}$, shown as Table 3.

The features selection result of water data set is $\{1, 2, 3, 4, 5, 6, 12, 13, 24, 25, 26, 27\}$, shown as Table 4.

The features selection result of NSL-KDD data set is $\{3, 5, 6, 23, 24, 29, 30, 31, 32, 33, 34, 35, 36, 37\}$, shown as Table 5.

The Modbus protocol is widely used in field network. Considering the structure of protocol, data features and information entropy, the above attributes were selected, as Table 3, Table 4 and Table 5 shown. Table 1 and Table 2 illustrate data composition, for example, the kinds of attacks. Table 6 shows the base condition of data set.

5.2 The Analysis of The Traffic Characteristics Map

Adopting traffic characteristics map method [19], traffic characteristics are extracted to produce traffic characteristics map.

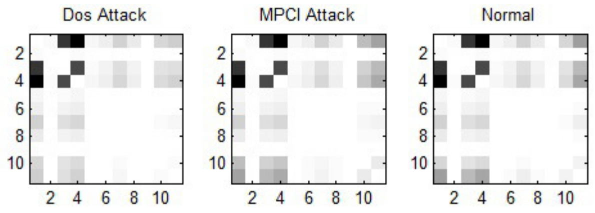


Figure 3: The traffic characteristic map of Dos, MPCI and normal traffic record in Data 1

According to the proposed method TCM-PH in Section 4, the experimental programs are realized in MATLAB. The testbed data and NSL-KDD data are chosen as the test data. The simulation results are shown in Figure 3. The subfigures express the different features in grey values. Dos attack and normal records are much different. The difference between MPCI and normal record is not much apparent. The reason is that the little difference in grey value can be recognized by TCM-PH method but not human vision.

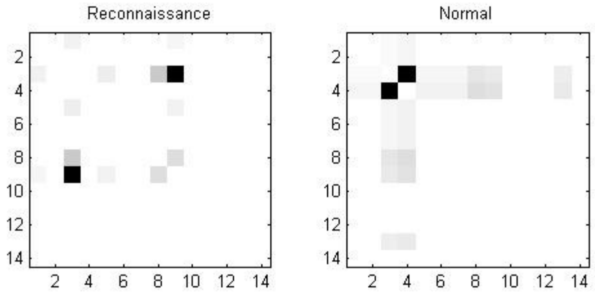


Figure 4: The traffic characteristic map of reconnaissance and normal traffic record in Data 3

In Figure 4, the difference between these two kinds traffic map is apparent. TCM-PH can also capture the features in the maps. The length of gas data is $L = 22$, and the length of water storage data is $L = 28$. The format of hash digest is binary strings. Table 7 describes the detection performance. TP is true positive rate, and FP is false positive rate.

In Figure 5, 5 kinds of records in the training data are shown. According to the results of the features selection method, the size of the map is 14×14. The difference between every map is obvious. These maps are the input data for the next operation.

Table 1: The kinds of Gas and Water data set

Label	Value	Description
Normal	0	Instance not part of an attack
NMRI	1	Naive malicious response injection attack
CMRI	2	Complex malicious response injection attack
MSCI	3	Malicious state command injection attack
MPCI	4	Malicious parameter command injection attack
MFCI	5	Malicious function command injection attack
Dos	6	Denial-of-service attack
Reconnaissance	7	Reconnaissance attack

Table 3: Attributes of gas data

Number	Attribute name	Description
1	*command_address*	Device ID in command packet
2	*response_address*	Device ID in response packet
3	*command_memory*	Memory start position in command packet
4	*response_memory*	Memory start position in response packet
5	*command_memory_count*	Number of memory bytes for R/W command
6	*response_memory_count*	Number of memory bytes for R/W response
12	*command_length*	Command packet length
13	*response_length*	Response packet length
24	CRC rate	CRC error rate
25	measurement	Pipeline pressure or water level
26	time	Time interval between two packets
27	result	Kinds of data

Table 4: Attributes of water data

Number	Attribute name	Description
1	*command_address*	Device ID in command packet
2	*response_address*	Device ID in response packet
3	*command_memory*	Memory start position in command packet
4	*response_memory*	Memory start position in response packet
5	*command_memory_count*	Number of memory bytes for R/W command
6	*response_memory_count*	Number of memory bytes for R/W response
10	*resp_fun*	Response function code
12	*command_length*	Command packet length
13	*response_length*	Response packet length
18	*control_model*	Automatic, manual or shutdown
20	pump-state	Compressor/pump state
21	CRC rate	CRC error rate
22	measurement	Pipeline pressure or water level
23	time	Time interval between two packets
24	result	Kinds of data

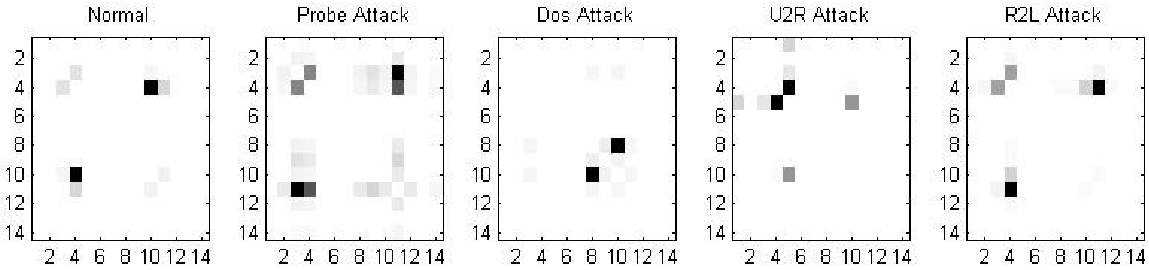


Figure 5: The traffic characteristic map of NSL-KDD train data set

Table 5: Attributes of NSL-KDD data

Number	Attribute name	Description
3	service	Service Type
5	*src_bytes*	The number of bits from source to destination
6	*dst_bytes*	The number of bits from destination to source
23	count	Number of connecting same hosts in past 2s
24	Srv count	Number of connecting same services in past 2s
29	same srv rate	Rate of same connecting service
30	diff srv rate	Rate of different connecting service
31	srv diff host rate	Rate of different connecting host
32	dst host count	Number of connecting same host
33	dst host srv count	Number of same host and same service
34	dst host same srv rate	Rate of same host and same service
35	dst host diff srv rate	Rate of different service in different host
36	dst host same src port rate	Rate of connecting host in same src port
37	dst host diff src port rate	Rate of connecting host in different src port
42	type	Kinds of data

Table 6: Experimental data composition

Name	Dimension	Normal Record Number	Abnormal Record Number	Attack Kinds
Data 1	2027 × 27	1732	295	$MPCI\&Dos$
Data 2	2844 × 27	2594	250	$MPCI\&Dos$
Data 3	23673 × 24	9554	14119	Reconnaissance
Data 4	1664 × 24	657	1007	Reconnaissance
NSL-KDD train	25192 × 42	13449	11743	$Dos\&Probe\&U2R\&R2L$
NSL-KDD test	22544 × 42	9711	12833	$Dos\&Probe\&U2R\&R2L$

Table 7: Rule set captured from Data 1 and Data 3.

Name	Normal Rule Set	Abnormal Rule Set	TP(mean)	FP(mean)
Data 1	75	103	0.9866	0.014
Data 3	76	67	0.9925	0.015
NSL-KDD train	471	535	0.9893	0.0012

5.3 Discrimination Experiments

The robustness and discrimination of TCM-PH are essential and vital for the abnormal intrusion detection. The robustness ensures that the same traffic record can produce same hash digest. The discrimination ensures that different and unknown attacks can map into different hash digests. The evaluation of discrimination is the false accepting rate (FAR) [8].

$$FRA = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\tau} \exp[\frac{-(x-\mu)^2}{2\sigma^2}] \qquad (12)$$

where μ is the mean of normal distribution and σ is the standard deviation. τ is the matching threshold.

In total, 143 different hash digests were taken to test the discrimination of TCM-PH algorithm. The total matching times is 10,153. Figure 6 is the normal distribution curve of the false accepting rate. The blue curve is coincided with the mean straight line. But, there are still some fluctuations.

The mean is 0.4991, and the theoretical standard deviation is 0.0418. And, the real standard deviation is

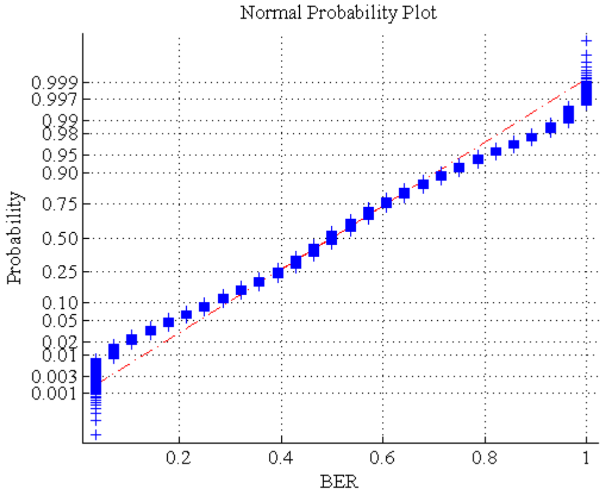


Figure 6: Normplot image of our scheme

0.1791. Both ends of the curve appear aggregation phenomenon. The aggregation in lower left quarter expresses that the discrimination of hash digest is very high. Another aggregation in upper right corner expresses that there indeed exist many analogical traffic characteristics maps, which is corresponding to the periodicity of traffic information.

Table 8: Comparing with FAR

Threshold τ	0.0357	0.02	0.01	0.005
FAR	0.0048	0.0037	0.0032	0.0029

When $\tau = 0.0357$, $FAR = 0.0048$. That is to say that there happen 4.8 false accepting intrusion attacks in 1000 traffic records, which meets the network detection request. Table 8 shows the correlations between FAR and τ.

The format of the hash digest is binary string. According to Equation (9), the hash distance is the normalized Hamming distance. It is also named as BER. We can think that every bit of hash digest is independent and identically distributed. Each bit can take the value at 0 or 1. The probability of these two values is equal. The probability is 0.5. The normalization Hamming distance obeys to normal distribution, which has 0.5 mean value and $\sigma = 0.5/\sqrt{N}$ standard deviation. When the attributes transform into binary string, the redundant information between attributes are kept. Therefore, the real standard deviation has little difference with the theoretical standard deviation. Figure 7 is the bit error rate (BER) colour histogram of discrimination of TCM-PH. The centre of BER distribution is close to 0.5, which is 0.4991. And standard deviation is 0.1791. The proposed algorithm has a good discrimination.

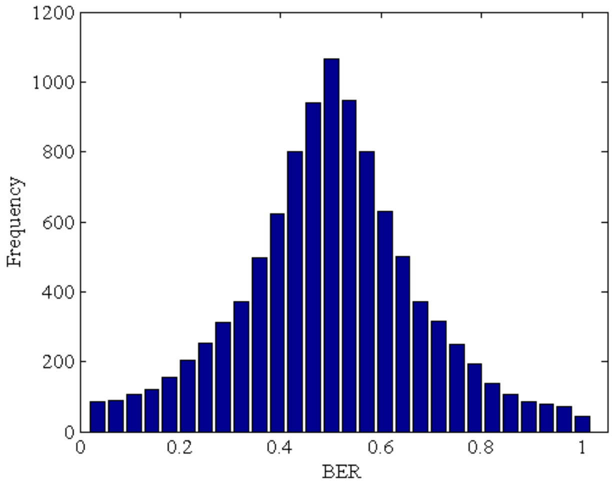


Figure 7: Colour histogram of BER

5.4 Algorithm Performance Analysis

The gas pipeline data, water storage data and NSL-KDD data set are adopted in experiments. The performance of the proposed method is shown in Table 9. The average TP of gas data is 0.986, and the corresponding average FP is 0.014. The average TP of water storage data is 0.9925, and the corresponding average FP is 0.015. The average TP of NSL-KDD is 0.9893, and the average FP is 0.0012. The TP of water storage data and NSL-KDD data are higher than gas pipeline data, which demonstrates that the precise detection needs more training data.

The number of training records is N_1 and the number of test records is N_2. The number of attributes is M. The number of normal hash digests is $nhash$ and the number of abnormal hash digests is $ahash$. The time complexity of TCM-PH algorithm is $O(N_1 + N_2)(M^2 + 5M)$ which is little bigger than MCA [19], and TP is 0.993. The FP of proposed method is minimum value. In [19], the complexity of MCA is $O(M^4)$. In [15], the complexity is $O((N_1+N_2)M^2)$. The complexity of the proposed method is better than Ref. [7, 16, 28].

6 Conclusions

In this paper, our study deal with three issues, transformation and features extraction of traffic characteristics map, intrusion detection rule matching problem and the proof of the robustness and discrimination of TCM-PH. By using traffic characteristics map technology, the text data are transformed into figure information. The features of the figure can be captured via perceptual hash features extraction method, which provides the new solutions from the figure features extraction point to deal with intrusion detection in industrial internet area. The three-level detection pattern adds the adaptability of our method. With this help, many unknown attacks can be recognized. The experimental results proved the robustness and discrimination of TCM-PH method, which provides theoretical support to our research. The experiments prove the feasibility of TCM-PH algorithm. The most vital result is that traffic characteristics map method provides network intrusion detection with new solutions.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61363078), the Natural Science Foundation of Gansu Province of China (No. 1310RJYA004), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (No. 201700005). The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

Table 9: Detection performance analysis

Method	**TP**	**FP**	**Time Complexity**
AP [7]	0.9436	0.08	$O(N_1 N_2 M^2)$
MSPCA [15]	0.9	0.2	$O((N_1 + N_2) M^2)$
PSO-SVM [16]	0.9583	-	$O(200 \times N_1 N_2)$
MCA [19]	0.993	0.018	$O(M^4)$
SVDD [28]	0.970	0.070	$O(N_1^2 N_2)$
TCM-PH of our scheme (gas data)	0.986	0.014	$O((N_1 + N_2)(M^2 + 5M))$
TCM-PH of our scheme (water data)	0.9925	0.015	$O((N_1 + N_2)(M^2 + 5M))$
TCM-PH of our scheme (NSL-KDD)	0.9893	0.0012	$O((N_1 + N_2)(M^2 + 5M))$

References

[1] M. El Boujnouni and M. Jedra, "New intrusion detection system based on support vector domain description with information gain metric," *International Journal of Network Security*, vol. 20, no. 1, pp. 25-34, 2018.

[2] N. Chen, H. D. Xiao, J. Zhu, J. J. Lin, Y. Wang, and W. H. Yuan, "Robust audio hashing scheme based on cochleogram and cross recurrence analysis," *Electronics Letters*, vol. 49, no. 1, pp. 7–8, 2013.

[3] R. H. Dong, D. F. Wu, Q. Y. Zhang and H. X. Duan, "Mutual information-based intrusion detection model for industrial internet," *International Journal of Network Security*, vol. 20, no. 1, pp. 131-140, 2018.

[4] N. Erez and A. Wool, "Control variable classification, modelling and anomaly detection in modbus/tcp scada systems," *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 59–70, 2015.

[5] N. Goldenberg and A. Wool, "Accurate modelling of modbus/tcp for intrusion detection in scada systems," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 63–75, 2013.

[6] ICS-CERT, "Monitor (ics-mm201612)," *https://ics-cert.us-cert.gov/monitors/ICS-MM201612*, November 2016.

[7] J. Jiang, Z. F. Wang, T. M. Chen, C. Zhu, and B. Chen, "Adaptive ap clustering algorithm and its application on intrusion detection," *Journal of Communication*, vol. 36, no. 11, pp. 119–126, 2015.

[8] Z. Jie, "A novel block-dct and pca based image perceptual hashing algorithm," *International Journal of Computer Science Issues*, vol. 10, no. 3, pp. 399–403, 2013.

[9] T. Morris and W. Gao, "Industrial control system traffic data sets for intrusion detection research," in *International Conference on Critical Infrastructure Protection*, pp. 65–78, Berlin, Heidelberg, March 2014.

[10] T. H. Morris, B. A. Jones, R. B. Vaughn, and Y. S. Dandass, "Deterministic intrusion detection rules for modbus protocols," in *46th Hawaii International Conference on System Sciences (HICSS)*, pp. 1773–1781, Wailea, Maui, HI, USA, January 2013.

[11] X. K. Mu, J. S. Wang, Y. F. Xue, and W. Huang, "Abnormal network traffic detection approach based on alive entropy," *Journal of Communication*, vol. 34, no. Z2, pp. 51–57, 2013.

[12] A. Nezarat, "Distributed intrusion detection system based on mixed cooperative and non-cooperative game theoretical model," *International Journal of Network Security*, vol. 20, no. 1, pp. 56-64, 2018.

[13] X. M. Niu and Y. H. Jiao, "An overview of perceptual hashing(in chinese)," *ACTA ELECTRONICA SINCA*, vol. 36, no. 7, pp. 1405–1411, 2008.

[14] M. Nouri, N. Farhangian, Z. Zeinolabedini, and M. Safarinia, "Conceptual authentication speech hashing base upon hypotrochoid graph," in *Sixth International Symposium on Telecommunications (IST)*, pp. 1136–1141, Tehran, Iran, November 2012.

[15] Y. K. Qian, M. Chen, L. X. Ye, F. Liu, S. Zhu, and H. Zhang, "Network-wide anomaly detection method based on multi-scale principal component analysis," *Journal of Software*, vol. 23, no. 2, pp. 361–377, 2012.

[16] W. L. Shang, S. S. Zhang, and M. Wan, "Modbus/tcp communication anomaly detection based on pso-svm," *Applied Mechanics and Materials*, vol. 490, pp. 1745–1753, 2014.

[17] S. N. Shirazi, S. A. Gouglidi, K. N. Syeda, S. Simpson, A. Mauthe, I. M. Stephanakis, and D. Hutchison, "Evaluation of anomaly detection techniques for scada communication resilience," in *Resilience Week (RWS)*, pp. 140–145, Chicago, IL, USA, August 2016.

[18] D. Stiawan, M. Y. B. Idris, A. H. Abdullah, and A. Mohammed, "Penetration testing and mitigation of vulnerabilities windows server," *International Journal of Network Security*, vol. 18, no. 3, pp. 501–513, 2016.

[19] Z. Tan, A. Jamdagni, and X. He, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE transactions on parallel and distributed systems*, vol. 25, no. 2, pp. 447–456, 2014.

[20] M. Tavallaee, E. Bagheri, W. Lu, , and A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pp. 1–6, Ottawa, ON, Canada, July 2009.

[21] W. Tylman, "Native support for modbus rtu protocol in snort intrusion detection system," *New Results in Dependability and Computer Systems*, vol. 224, pp. 479–487, 2013.

[22] W. Tylman, "Scada intrusion detection based on modelling of allowed communication patterns," *New Results in Dependability and Computer Systems*, vol. 224, pp. 489–500, 2013.

[23] A. G. Voyiatzis, K. Katsigiannis, and S. Koubias, "A modbus/tcp fuzzer for testing internetworked industrial systems," in *20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pp. 1–6, Luxembourg, September 2015.

[24] M. Wan, W. L. Shang, and P. Zeng, "Anomaly detection approach based on function code traffic by using cusum algorithm," in *4th National Conference on Electrical, Electronics and Computer Engineering (NCEECE)*, pp. 12–13, Xian, China, December 2015.

[25] W. Wang, Y. He, J. Liu, and S. Gombault, "Constructing important features from massive network traffic for lightweight intrusion detection," *IET Information Security*, vol. 9, no. 6, pp. 374–379, 2015.

[26] Q. Xiong, H. Liu, Y. Xu, H. Rao, S. Yi, B. Zhang, W. Jia, and H.Deng, "A vulnerability detecting method for modbus-tcp based on smart fuzzing mechanism," in *International Conference on Electro/Information Technology (EIT)*, pp. 404–409, Dekalb, IL, USA, May 2015.

[27] Y. Zhao, Q. Wang, Y. Z. Huang, W. Qing, and Z. Sheng, "Collaborative visual analytics for network traffic time-series data with multiple views," *Journal of Software*, vol. 27, no. 5, pp. 1118–1198, 2016.

[28] L. M. Zheng, P. Zou, Y. Jia, and W. H. Hang, "How to extract and train the classifier in traffic anomaly detection system," *Chinese journal of computer*, vol. 25, no. 4, pp. 719–729, 2012.

Biography

Dong Ruihong Vice researcher, worked at school of computer and communication in Lanzhou university of technology. His research interests include network and information security, information hiding and steganalysis analysis, computer network.

Wu Dongfang In 2015, Wu Dongfang obtained his bachelor of engineering degree from Northwest University for Nationalities. Currently, he is studying for his master's degree at Lanzhou University of Technology. His research focuses on the industrial control network security.

Zhang Qiuyu Researcher/PhD supervisor, graduated from Gansu University of Technology in 1986, and then worked at school of computer and communication in Lanzhou University of Technology. He is vice dean of Gansu manufacturing information engineering research centre, a CCF senior member, a member of IEEE and ACM. His research interests include network and information security, information hiding and steganalysis, multimedia communication technology.

Zhang Tao He is studying for his master's degree at Lanzhou University of Technology. His research focuses on the network and information security.

New Generic Design to Expedite Asymmetric Cryptosystems Using Three-levels of Parallelism

Mohamed Rasslan[1], Ghada Elkabbany[2], and Heba Aslan[3]

(Corresponding author: Mohamed Raslan)

Informatics Dept., Electronics Research Institute
National Research Center building, El Tahrir, St. Dokki, Giza, Egypt
(Email: mohamedraslan@eri.sci.eg)

(Received Jan. 15, 2017; revised and accepted May 16 & June 20, 2017)

Abstract

Public key cryptosystems depend on complex and time consuming arithmetic operations. Public key cryptosystems require modular operations over large numbers or finite fields. Researchers are working on improving the performance of asymmetric cryptosystems while maintaining the security of the cryptographic algorithms. Parallel computing is the most promising solution to improve the computing power and speed-up these arithmetic operations. In this paper, we propose a generic model to execute any encryption algorithm through a parallel-pipelined design. The proposed design is twofold. First, we make use of a number of processors to execute different encryption/decryption steps in parallel. Secondly, complex arithmetic operations could be divided into small simple arithmetic operations that are executed in parallel. Simulation experiments show that parallel implementations of the aggregated signcryption protocol (as a case study) outperforms the sequential performance. The average values of improvement ranges from 47.5% to 80.4% for different number of processors.

Keywords: Cryptography; Modular Operations; Parallelization; Public Key Cryptosystems

1 Introduction

Security is a serious issue when it comes to carry information over non-secure channels. In the era of Internet of Things, sensitive information requires secure transmission over the Internet. Cryptography is one of the most popular techniques that provide security and integrity to sensitive information over adversarial communication channels. Cryptographic algorithms are classified into two distinct categories: Symmetric and Asymmetric Key algorithms. Symmetric Key algorithms use a single key in order to encrypt, as well as, decrypt the data. Asymmetric key algorithms use two different keys in order to perform encryption and decryption. To carry out secure communication between contending parties, the receiver generates the public key and the corresponding private key. Then, the receiver shares the public key with the sender using a certificate authority. The sender encrypts the message using the public key and sends the cipher text to the receiver. The receiver decrypts the cipher text using corresponding private key and recovers the message. There are many public key based algorithms [28] (i.e. RSA, Digital Signature Algorithm, and Diffie-Hellman Key Exchange Algorithm.) Cryptographic algorithms are sequential algorithms. A single processor executes instructions one by one in sequence.

Asymmetric algorithms make use of modular arithmetic that requires complex computation for very large integers. Sequential implementation of complex asymmetric algorithms degrades the performance of encryption and decryption. Moreover, these algorithms demand huge memory size and have high power consumption. Parallelization techniques reduce the power consumption and achieve high performance in terms of execution time. Parallelization techniques use multi-core processors for efficient execution of instructions. In a parallel process, the processing is broken down into parts that are executed concurrently on different CPUs. Security algorithms can be implemented in parallel after dividing the algorithm into specific parts that can be executed on multi-core processor in order to enhance the performance.

Multimedia applications, such as, video on demand, distance learning, interactive e-Commerce, and TV channels delivery via Internet require secure communication over the Internet in order to transfer data. Cryptography protects data to achieve security services. Modern multimedia applications are real time applications that have concerns regarding delay introduced by cryptography. To expedite cryptographic computations, there are two approaches. First, design faster cryptographic algorithms. It is not a practical option, because standard cryptographic algorithms take time to be developed and tested. Moreover, performance of cryptographic algorithm depends on number of rounds or by the size of the message. Secondly, to expedite cryptographic computations, we can deploy a parallel cryptographic system. The parallel encryption and signing was introduced by [9, 25, 36].

In this paper, we propose a generic model to execute any encryption algorithm (symmetric or public) through a parallel-pipelined design. We accelerate the cryptographic algorithms through parallel-pipelined design. We make use of *"M"* processing elements (PEs) to execute different encryption/decryption steps in parallel. Then, complex arithmetic operations could be divided into small simple arithmetic operations that are executed in parallel (load-balancing level). Simulation experiments show that parallel implementations of the aggregated signcryption protocol (as a case study) outperforms the sequential performance. The average values of improvement ranges from 47.5% to 80.4% for different number of processors.

This paper is structured as follows. Section 2 puts forward background and related work. Next, cryptographic arithmetic model is discussed in Section 3. The description of expedited asymmetric cryptosystems is presented in Section 4. Section 5 concludes the work.

2 Background and Related Work

Parallel techniques are used to accelerate several cryptographic algorithms (symmetric [10, 11, 12, 13, 29] and asymmetric [1, 3, 5, 7, 8, 18, 21, 22, 24, 27, 31, 33, 34, 35]). Practical asymmetric (public) cryptography requires modular operations over large numbers that is considered as computationally exhaustive process. Many researchers have done work in order to expedite the performance of asymmetric cryptography.

RSA is one of the most popular public key cryptography based algorithm. It is based on the mathematical scheme of factorization of very large integers which is a compute-intensive process and takes very long time. Several scientists used parallel computing to speed up the RSA algorithm. Saxena and Kapoor [31] presented a survey of various parallel implementations of RSA algorithm involving variety of hardware and software implementations.

Bajard and Laurent [1] presented a full implementation of RSA that has an efficient hardware implementation. It is sequential in nature but gives high throughput. Rawat and Walfish [27] presented a parallel signcryption standard using RSA with Probabilistic Signature and Encryption Padding (PSEP). Ciet *et al.* [7] presented a parallel FPGA implementation of RSA with Residue Number Systems (RNS). Tang *et al.* [33] presented a modular exponentiation technique using parallel multipliers. Wu *et al.* [34] offered a fast parallel technique has a speedup of 1.06 to 2.75. Liu *et al.* [22] proposed a high performance VLSI implementation of the RSA algorithm. Chang *et al.* [5] presented a fast parallel molecular algorithm for DNA-Based computation: factoring integers. Lin *et al.* [21] presented an efficient parallel RSA decryption algorithm for many-core Graphics Processing Unit (GPUs) with Compute Unified Device Architecture (CUDA.) Zhang *et al.* [35] presented a comparison and analysis of General-Purpose computing on Graphics Processing Units (GPGPU) and parallel computing on multicore CPU. Damrudi and Ithnin [8] presented a parallel RSA encryption based on tree architecture. Mahajan and Singh [24] presented an analysis of RSA algorithm using GPU programming. Mahajan and Singh described that the GPU as a coprocessor of CPU can be used to implement massive parallelism. Mahajan and Singh designed parallel RSA algorithm for GPU using CUDA framework and tested for both small and large prime numbers.

Elliptic Curve Cryptosystems (ECC) is used among the cryptographic community for their relatively better security and ease of implementation [3]. Several researchers used parallel computing to speed up the ECC algorithm. Basu [3] presented a parallel algorithm for elliptic curve cryptography (ECC). His simulation studies had been performed by implementing the parallel algorithm on a multi-core architecture (upto 8 cores). Hossain *et al.* [18] proposed a parallel architecture for fast hardware implementation of ECPM. It has been implemented over the binary field, and supported two Koblitz and random curves for the key sizes 233 and 163 bits.

In this work, we propose a generic parallel-pipelined design to speed up different encryption algorithms (symmetric or public). In the next section, cryptographic arithmetic model is discussed.

3 Cryptographic Arithmetic Model

Security services are generally classified into six components: confidentiality, data integrity, authentication, authorization, non-repudiation, and accountability. To achieve those issues different encryption/decryption protocols have been proposed, all those protocols are based on complicated mathematical operations. These operations can be divided into modulo operations (such as: modular addition, modular multiplication, modular subtraction, modular exponential, multiplicative inverse, and etc.), logical operations (such as: ANDing, inverse AND (NAND), ORing, XORing, Shift left, Shift right, Rotate left, Rotate right).

Some encryption techniques need more than one step (encryption or decryption) to complete the message encryption such as Aggregated Signcryption [4, 6, 15], Sign-Encrypt-Sign, Encrypt-Sign-Encrypt [26], and Schnorr Signcryption [30, 32]. The total execution/sequential time for one message (Tmess) can be calculated as follows:

$$Tmess = TKG + \sum_{l=1}^{\gamma} T_{step_l} \tag{1}$$

Where *TKG* is the key generation time, *Tstep* is the time needed to execute encryption or decryption operation in one step, and "γ" is the number of steps.

$$TKG = \sum_{l=1}^{key} T_{kg_l} \tag{2}$$

where, key is the number of keys and

$$Tstep = \sum_{i=1}^{K} T_{stagei} \tag{3}$$

where K is the number of the encryption/decryption stages.

To calculate the total time for any encryption or decryption stage T_{stagei}, we assume that this stage needs *ai,*1 modular additional operations, *ai,*2 modular subtraction operations, *ai,*3 modular multiplication operations, *ai,*4 modular exponential operations, *ai,*5 modular multiplication inverse operations, and *ai,*6 logical operations (may include: AND, OR, XOR, NAND, Rotate left, Rotate right and etc.). T_{stagei} is calculated as follows:

$$\begin{aligned} T_{stagei} = & \quad a_{1,i}t_{mod-add} + a_{2,i}t_{mod-sub} \\ + & \quad a_{3,i}t_{mod-mul} + a_{4,i}t_{mod-ex} \\ + & \quad a_{5,i}t_{mul-inv} + a_{6,i}t_{\log} \end{aligned} \tag{4}$$

Then,

$$Tstep = \sum_{i,j}^{K,f} a_{j,i}t_{operation(op)} \tag{5}$$

where operations ∈{Modular addition (mod-add), Modular subtraction (mod-sub), Modular Multiplication (mod-mul), Modular exponential (mod-ex), Modular multiplication inverse (mul-inv) and others logical operations}, f = |Operation|, 1≤ j ≤ f, and 1≤ i ≤ K.

Figure 1 presented a general form of an encryption protocol. Assuming that, t_{add} is the time needed for computing simple addition operation, t_{sub} is the time needed for completing simple subtraction operation, t_{mul} is the time needed to execute simple multiplication operation, t_{div} is the time needed to perform simple division operation, and $t_{sub} = t_{add}$.

Modular addition can be calculated as the summation of addition and modulo operations. Using Barret algorithm [2], modulo operation needs one normal/simple multiplication, one normal division, and one normal subtraction. Then, the time needed to compute modular addition operation (which can be divided into one addition, one subtraction, one multiplication and one division), is $t_{mod-add}$ and is given by:

$$\begin{aligned} t_{mod-add} &= t_{add} + t_{modulo} \\ &= 2t_{add} + t_{div} + t_{mul} \end{aligned} \tag{6}$$

On the other hand, the time needed to compute modular subtraction operation which can be done by simple subtraction followed by modular addition; is $t_{mod-sub}$ and calculated as:

$$t_{mod-sub} = 3t_{add} + t_{div} + t_{mul} \tag{7}$$

Moreover, the time needed to compute modular multiplication operation, which can be divided into three simple multiplication operation and one simple addition operation [16] is $t_{mod-mul}$.

$$t_{mod-mul} = t_{add} + 3t_{mul} \tag{8}$$

Furthermore, to compute modular exponentiation operation using Indian algorithm [20], it needs approximately $\left(\frac{3b}{2}\right)$ modular multiplications for an exponent "e" of "b-bit", assuming the exponent contains approximately 50% ones/zeros. That is to say, a modular exponentiation "$Y = X^e modn$" is performed by successive modular multiplication and the time needed to compute modular exponentiation operation t_{mod-ex} can be calculated as follows:

$$\begin{aligned} t_{mod-ex} &= \tfrac{3b}{2}t_{mod-mul} \\ &= \tfrac{3b}{2}(t_{add} + 3t_{mul}) \end{aligned} \tag{9}$$

On the other hand, modular multiplication inverse is executed using Euler's theorem [19], therefore, it could be done using modular exponential, and then the time needed to compute multiplication inverse operation can be computed as follows:

$$\begin{aligned} t_{mul-inv} &= t_{mod-ex} \\ &= \tfrac{3b}{2}(t_{add} + 3t_{mul}) \end{aligned} \tag{10}$$

From Equation (1) and Equation (4), the time needed to encrypt/decrypt this stream of data (N messages) on one processing element (Ts) is given by the following equation:

$$Ts = N * (\sum_{l=1}^{key} T_{kg_l} + \gamma \sum_{i,j}^{n,f} a_{j,i}t_{operation(op)}) \tag{11}$$

In case of all stages have the same operations; the time needed to encrypt/decrypt this stream of data on one processing element (PE) is given by:

$$Ts = N * (\sum_{l=1}^{key} T_{kg_l} + \gamma * \sum_{j}^{f} a_j t_{operation(op)}) \tag{12}$$

where $\sum_j^f a_j t_{operation}$ is time needed to execute any stage$_i$, $1 \leq i \leq$ K.

Due to the nature of most security algorithms, which are characterized by repeating the same function for several messages, different levels of parallelism can be used. This will improve the system utilization and throughput. In the next section, we use parallel and pipeline techniques to speed up these protocols.

4 Expediting Asymmetric Cryptosystems

Parallel systems, which emphasize parallel processing, are the most favorable architectures to increase the computing power and achieve speedup. Parallel processing continues to hold the promise of the solution of more complex problems, by connecting a number of powerful processing elements (PEs) together into a single system. These connected processors cooperate to solve a single problem that exceeds the ability of any one of the processing elements (PEs). That is to say, parallel computing is the simultaneous execution of the same task (split up) on multiple processing elements (PEs) in order to obtain faster results. The idea is based on the fact that the process of solving problem can be divided into small tasks, which

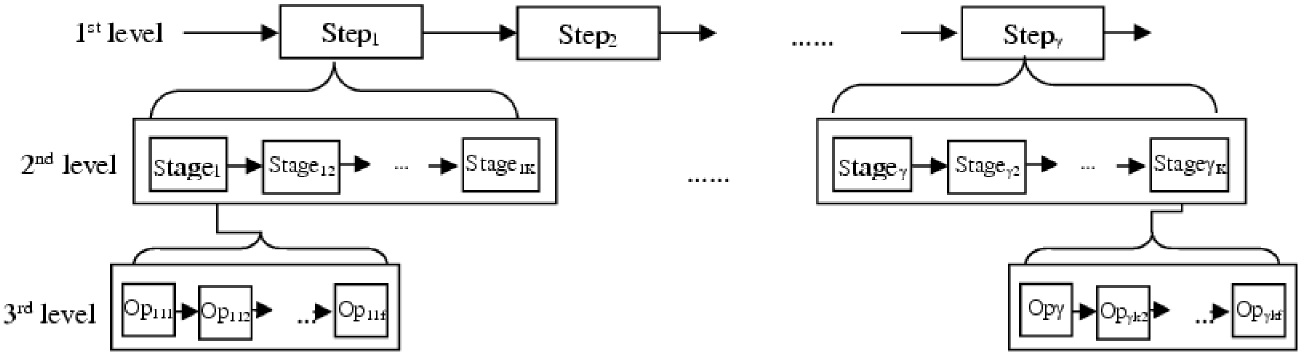


Figure 1: General form of an encryption/decryption protocol

may be carried out simultaneously with some coordination.

In our work, we will not concentrate on parallelizing the key generation process, since; it is done for several messages. However, we recommend making use of the parallel architecture in its implementation. From Equation (1), *Ts* for '*N*' messages is calculated as follows:

$$TNs = N * \sum_{l=1}^{\gamma} T_{step_l} \tag{13}$$

Our proposed design is composed of three levels of parallelism: *first* different steps are pipelined, *then*, different stages inside each step are pipelined, and *finally*, load is balanced between PEs.

4.1 First Level: Pipelining Encryption/Decryption Processes

Pipelining is a kind of parallel computing that increases system performance by taking the advantage of the intrinsic parallelism by breaking a process into sub-processes executed by different PEs with inputs streaming through. There are two pipeline levels; the first one is pipelining different steps, while the second one is through pipelining the stages of each encryption/decryption process individually.

In this section, we will discuss how to pipeline different steps on a computer cluster equipped with 'γ' homogenous clusters 'C', where $C = \{C1,C2,..., C\gamma\}$. Assuming that the time needed to execute step$_l$ on cluster C_l is *Tstepl* the parallel/pipelined time *Tpar* is as follows:

$$T_{par} = N\ {*}T_{step_{max}} + (\sum_{l=1}^{\gamma} T_{step_l} - T_{step_{max}}) \tag{14}$$

where, $T_{step_{max}}$ is the needed time to execute the latest encryption/decryption operation. From Equations (13) and (14), the improvement in the execution time can be executed as follows:

$$\frac{T_{Ns} - T_{par}}{T_{Ns}} = \frac{(N-1)*(\sum_{l=1}^{\gamma} T_{step_l} - T_{step_{max}})}{N*\sum_{l=1}^{\gamma} T_{step_l}} \tag{15}$$

In case of repeated processes (all processes need the same execution time). Therefore, $T_{par}$is given by:

$$T_{par} = (N + \gamma - 1)\ {*}T_{step_l} \tag{16}$$

From Equation (13) and Equation (16), the improvement in the execution time is executed as follows:

$$\frac{T_{Ns} - T_{par}}{T_{Ns}} = \frac{(N-1)*(\gamma-1)}{N*\gamma} \tag{17}$$

4.2 Second Level: Pipelining Different Stages of an Encryption/Decryption Operation

In this section, different stages of each step are executed in a stream of PEs in a pipelined manner separately. As mentioned at the previous section, each encryption/decryption process can be executed using a separate cluster, and assuming that inside each cluster there are 'M homogenous processing elements (PEs), where for cluster C$_l$, PE$_l$={PE$_{l,1}$,PE$_{l,2}$,..., PE$_{l,M}$}.That is to say inside each cluster 'M' PEs can cooperate to accelerate each process separately. To execute any encryption/decryption process using 'M' processing elements, there are three cases: i) number of PEs is smaller than the number of stages (M<K), ii) number of PEs equals the number of stages (M = K), and iii) number of PEs is greater than the number of stages (M > K).

4.2.1 Number of PEs (M) is Smaller than the Number of Stages (K)

In this case, to achieve the optimum system effectiveness one or more consecutive stages can be executed by each PE. In other words, the first PE$_1$ gives its output to the subsequent one PE$_2$. This is repeated for the different PEs. Assuming that the time needed to perform stage number 'i' is T$_{stagei}$, and from Equation (2), the total sequential time to execute 'N' message is:

$$T_S = N * \sum_{i=1}^{K} T_{stage_i} \tag{18}$$

Each PE$_l$needs T$_{PEl}$ to compute its assigned tasks, and the maximum time is:

$$T_{\max} = \max_{l=1}^{M} T_{PEl} \tag{19}$$

Then the parallel time $Tpar$ is calculated as follows:

$$T_{par} = \sum_{i=1}^{K} T_{stage_i} + (N-1)T_{\max} \quad (20)$$

From Equation (18), and Equation (20), the improvement in the execution time can be calculated as follows:

$$\frac{T_s - T_{par}}{T_s} = \frac{(N-1)*(\sum_{i=1}^{K} T_{stage_i} - T_{\max})}{(N*\sum_{i=1}^{K} T_{stage_i})} \quad (21)$$

In case of repeated stages, the output of each stage$i$ is shifted to stage$i$+1 for all stages at the same time, and from Equation (13), the total sequential time can be calculated as follows:

$$Ts = N * K * T_{stage} \quad (22)$$

The parallel/pipelined time $Tpar$ is calculated for the following two cases:

1) ***First***, M divides K (K/M= integer), where each PE executes the same number of stages, then the execution time $Tpar$ is calculated as follows:

$$T_{Par} = (K + (N-1) * (\frac{K}{M})) * T_{stage} \quad (23)$$

The improvement in the execution time; which is achieved through using 'M' PEs, is given by:

$$\frac{T_s - T_{par}}{T_s} = \frac{(N*K) - ((N-1)*(\frac{K}{M}) + K)}{(N*K)} \quad (24)$$

2) ***Second,*** M does not divide K ($K/M \neq$ integer). For the first ($K - \lfloor K/M \rfloor \times M$) PEs, each PE calculates$\lceil K/M \rceil$stages, and for the remaining PEs, each PE executes$\lfloor K/M \rfloor$stages. $T_{Par}$is given by:

$$T_{par} = (K + (N - \mathit{1})*(|\frac{K}{M}| + \mathit{1}))* \mathrm{T}_{stage} \quad (25)$$

From Equations (18) and (25), the improvement in the execution time can be calculates as follows:

$$\frac{T_s - T_{par}}{T_s} = \frac{(N*K) - ((N-1)*(|\frac{K}{M}|+1) + K)}{(N*K)} \quad (26)$$

4.2.2 Number of PEs (M) Equals to the Number of Stages (K)

In this case, the number of PEs equals to the number of stages to be executed (M=K). The total time to compute 'N' messages in parallel is given by the following equation:

$$T_{par} = (N-1) * T_{stage_{max}} + \sum_{i=\mathit{1}}^{K} T_{stage_i} \quad (27)$$

where $T_{stage_{max}}$ is the time needed to execute the latest stage, from Equations (3) and (27), the improvement in the execution time can be calculated as follows:

$$\frac{T_s - T_{par}}{T_s} = \frac{(N-1)*(\sum_{i=\mathit{1}}^{K} T_{stage_i} - T_{stage_{max}})}{N*\sum_{i=\mathit{1}}^{K} T_{stage_i}} \quad (28)$$

In case of repeated stages, the total time to compute 'N' messages in parallel is given by the following equation:

$$T_{par} = (N + K - 1) * T_{stage} \quad (29)$$

From Equations (18) and (29), the improvement in the execution time can be calculated as follows:

$$\frac{T_s - T_{par}}{T_s} = \frac{(N-1)*(K-1)}{(N*K)} \quad (30)$$

4.2.3 Number of PEs(M) is Greater Than the Number of Stages (K)

In this case, 'K' PEs are needed for computing 'K' stages and the remaining '*M-K'* PEs are idle. This leads to load imbalance, to avoid load imbalance, more than one PE can work together to compute different operations. In other words, parallelism is accomplished in each stage's operation level as explained in the next section.

4.3 Third Level: Parallelization Inside Individual Stages (Load Balancing)

Parallelism is implemented in each stage's operation level. This is done in the operation/instruction level, where complicated operations could be reduced into simple operations in order to be executed in parallel. This combination will improve the system utilization and throughput. As assumed in Section 3.1, we simplify each stage into multiple operations as shown in Equation (4). Each encryption/decryption stage$_i$ needs *ai,1* modular additional operations, *ai,2* modular subtraction operations, *ai,3*modular multiplication operations*, ai,4* modular exponential operations, *ai,5* modular multiplication inverse operations, and *ai,6* logical operations.

The time to carry out logical operations is relatively very small compared to the modular operations. Hence, it will be neglected in our calculations. On the other hand, we divide the arithmetic operations into low time consuming arithmetic operations (modular additional, modular subtraction), and highly time consuming (modular exponential, modular multiplication, and modular multiplication inverse). For the low level time consuming operations there is one PE who executes the calculations. That is to say, it will be performed sequentially. While for the highly time consuming operations, we reduce all those operations into modular multiplications. As mentioned at Equation (9), modular exponential operation needs approximately (3b/2) modular multiplications. Similarly, modular inverse multiplication operation needs approximately (3b/2) modular multiplications as mentioned at Equation (10). Therefore, the efficient execution of the modular multiplication is the key to improve the performance. Then, our objective is to reduce the modular multiplication time. This can be achieved by incorporating the idle PEs in the computing of each modular multiplication operation.

Assuming that each stage has at least one modular multiplication, the modular multiplication operation can be divided into three simple multiplications and one addition (as mentioned before) and could be performed in parallel using three processing elements (PEs).

Case 1: M =3K, two idle PEs can help each overloaded PEs to execute the modular multiplication operations, and the total parallel time can be decreased to approximately one-third of the time needed to execute the latest stage and calculated as follows:

$$T_{par} = \sum_{i=1}^{K} T_{stage_i} + (N-1)\frac{T_{\max}}{3} \quad (31)$$

Case 2: $M<3K$, the overloaded PEs must be arranged in an ascending manner, and the idle PEs must help the first $\left(\frac{M-K}{2}\right)$ overloaded PEs, and the total parallel time can be calculated as follows:

1) $\left(\frac{M-K}{2}\right) < 3$

$$T_{par} = \sum_{i=1}^{K} T_{stage_i} + (N-1)T_{ov} \qquad (32)$$

where 'T_{ov}': is the time needed by PE number $\left(\left|\frac{M-K}{2}\right|+1\right)$(at the queue) to execute its task.

2) $\left(\frac{M-K}{2}\right) \geq 3$

$$T_{par} = \sum_{i=1}^{K} T_{stage_i} + (N-1)T_{ov} \qquad (33)$$

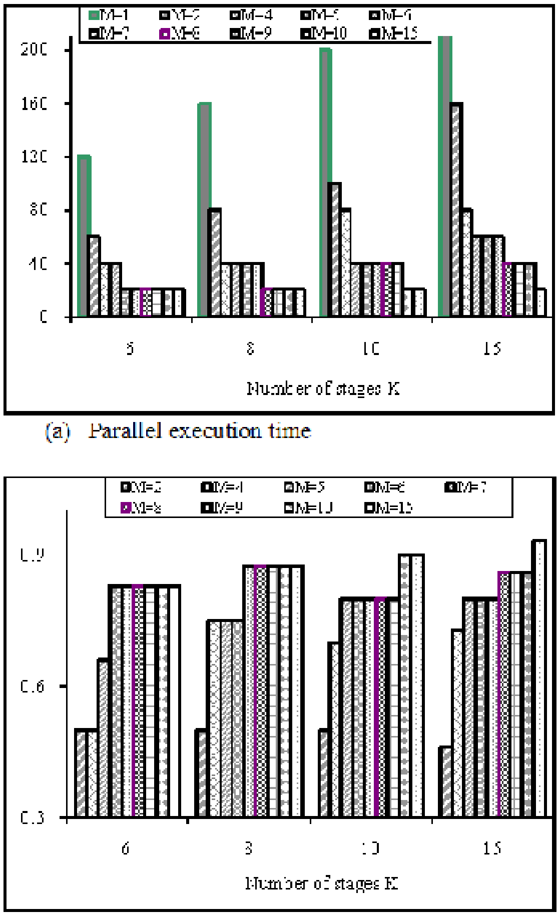
where 'T_{ov}': is the time needed by PE number $\left|\frac{M-K}{2}\right|$(at the queue) to execute its task.

4.4 Case Study: Aggregated Signcryption

In computer security, digital certificates are verified using a chain of trust. The trust anchor for the digital certificate is the root certificate authority (CA). The certificate hierarchy is a structure of certificates that allows individuals to verify the validity of a certificate's issuer. Certificates are issued and signed by certificates that reside higher in the certificate hierarchy, hence the validity and trustworthiness of a given certificate is determined by the corresponding validity of the certificate that signed it. The chain of trust of a certificate chain is an ordered list of certificates, containing an end-user subscriber certificate and intermediate certificates (that represents the intermediate CA), that enables the receiver to verify that the sender and all intermediates certificates are trustworthy.

Aggregate signature scheme [4, 6, 15] is a digital signature that supports aggregation. It is a single short string that convinces any verifier that, for all $1 \leq i \leq N$, signer S_i signed message $Mess_i$, where the N signers and N messages may all be distinct. The main motivation of aggregate signatures is compactness. That is to say, aggregate signatures are useful for reducing the size of certificate chains (by aggregating all signatures in the chain) and for reducing message size in secure routing protocols. In addition, aggregate signatures perform verifiably encrypted signatures. Such signatures enable the verifier to test that a given ciphertext C is the encryption of a signature on a given message '*Mess*' [4]. Bonds *et al.* [4] introduced the concept of an aggregate signature which is based on bilinear pairing, and gave several applications for aggregate signatures. In case of signers is ordered, the aggregate signature is computed by having each signer, in turn, add his signature to it [23]. In this section, we implement our proposed design on this scheme.

$$e(\sigma, g_2) = e\left(\prod_i h_i, v_i\right) = \prod_i e(h_i, v_i) \qquad (34)$$



(a) Parallel execution time

(b) Improvement degree

Figure 2: The system performance when a member of messages $N = 20$, for different values of $K = 6, 8, 10$, and 15

Here '$\gamma = 3$' and from Equation (1), Ts for 'N' messages is calculated as follows:

$$T_{Ns} = \sum_{l=1}^{3} T_{step_l} \qquad (35)$$

In order to simplify the calculations, we will neglect the time required to calculate Steps 2 and 3 as they are small compared to the time required to perform bilinear mapping operation (Step 1), In addition, we assume that Step 1 is done in 'K' stages, and all stages have the same execution time regardless the key or message length. From Equation (22) the total sequential time can be calculated as:

$$Ts = N * K * T_{stage} \qquad (36)$$

In the following paragraphs, we illustrate the three cases performing the aggregated signcryption.

Case 1: Distinct messages (N) with multiple signers $M < K$. There are three cases:

1) $K/M =$ integer, the parallel execution time and its improvement over sequential time is given by

:

$$T_{Par} = (N * (\frac{K}{M})) * T_{stage} \quad (37)$$

$$\frac{T_s - T_{par}}{T_s} = \frac{(N*K)-(N*(\frac{K}{M}))}{(N*K)} \quad (38)$$

2) $K/M \neq$ integer, the parallel execution time and its improvement over sequential time is given by :

$$T_{par} = (N*(|\frac{K}{M}| + 1))* \mathrm{T}_{stage} \quad (39)$$

$$\frac{T_s - T_{par}}{T_s} = \frac{(N*K)-(N*(|\frac{K}{M}|+1))}{(N*K)} \quad (40)$$

3) M=K, the execution time and its improvement over the sequential time can be calculated as shown:

$$T_{Par} = N * T_{stage} \quad (41)$$

$$\frac{T_s - T_{par}}{T_s} = \frac{(N*M)-(N)}{N*M} = \frac{M-1}{M} \quad (42)$$

Figure 2 presents the performance of the proposed model compared to the performance prior to parallelization for N = 20, and 'M ' ranges from 1 to 10 for different number of signers 'K' = 6,8,10, and 15 respectively.

1) Figure (2-a) demonstrates the total execution time (parallel time in terms of T_{stage}) for different values of 'K'. For different values of 'K', as the number of PEs increases, the total execution time decreases irrespective of the value of 'K'. When the nu mber of PEs is increased than ($K/2$), the execution time will be stabilized until the number of PEs reaches the number of signers 'K'. When M=K, a significant decrease in time occurs. Increasing the number of PEs than the number of stages 'K' leads to load imbalance. Consequently, the system's efficiency will decrease. Therefore, the number of PEs must not exceed a certain number which is called system's saturation. That is to say the saturation occurs when the number of PEs equals 'K'.

2) Figure (2-b) illustrates the degree of improvement of the proposed model compared to prior to parallelization. For different values of 'K', such as previously observed (execution time), as the number of PEs increases, the improvement degree increases irrespective of the value of 'K'. When the number of PEs is increased than ($K/2$), the improvement degree will be saturated until the number of PEs reaches the number of signers 'K'. When M=K, a considerable improvement occurs. For different values of 'K', the average values of improvement are 50%, 66.6%, 75.5%, 79.9% and 85.4% for M= 2, 3, 4, 6 and 8 respectively.

Case 2: Distinct messages (N) with ordered signatures.

In this case, different stages of Step 1 are executed in a stream of PEs in a pipelined manner separately. As mentioned at the previous section. We assumed repeated stages and we have two cases:

1) ***M<K***, and as mentioned above there are two cases:

   a. K/M= integer, the execution time and its improvement over the sequential is given by Equations (23) - (24).

   b. $K/M \neq$ integer the execution time and its improvement over the sequential is given by Equations (25) - (26).

2) M=K, execution time and its improvement over the sequential is given by Equations (29) - (30).

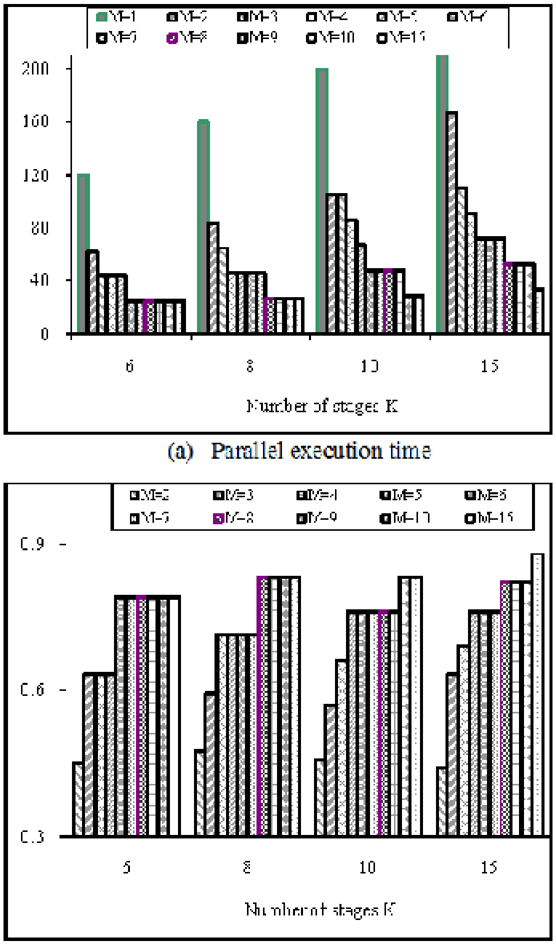
Figure 3 shows the performance of the proposed model with respect to the performance prior to pipelining for N = 20, and M = 1,2,3,4,5,6,7,8,9 and 10 for different number of signers 'K' = 6,8,10, and 15 correspondingly.

Figure (3-a) summarizes the total parallel time (in terms of T_{stage}) for different values of 'K'. As the number of PEs increases, the total execution time decreases irrespective of the value of 'K'. When the number of PEs is increased than ($K/2$), the execution time will be stabilized until the number of PEs reaches the number of signers 'K'. When M=K, a significant decrease in time occurs. When the number of PEs is increased than 'K', load imbalance will occur. Consequently, the system's efficiency will decrease. Therefore, the number of PEs must not exceed a certain number which is called system's saturation. That is to say the saturation occurs when the number of PEs equals 'K'.

Figure (3-b) illustrates the degree of improvement of the proposed model compared to prior to parallelization. For different values of 'K', as the number of PEs increases, the improvement degree increases irrespective of the value of 'K'. Such as previously observed (execution time), when the number of PEs is increased than ($K/2$), the improvement degree will be saturated until the number of PEs reaches the number of signers 'K'. When M=K, a considerable improvement occurs. For different values of 'K', the average values of improvement are 47.5%, 60.6%, 67.3%, 71.5% and 80.4% for M= 2, 3, 4, 6 and 8 respectively.

From Figures 2 and 3, we can deduce that as increasing the number of 'K', the execution time increases. This is due to the fact that the computation time increases as increasing the number of signers. This shows the advantage of using a multiprocessor system in enhancing the system performance.

Case 3: Single message with distinct signers.



(a) Parallel execution time

(b) Improvement degree

Figure 3: The performance of the proposed model with respect to the performance prior to pipelining for $N = 20$, and $M = 1, 2, \cdots, 10$ for different number of signers $K = 6, 8, 10$, and 15

One or more PEs co-operates to executee $(h, \prod_i v_i)$. In this case the parallelization is done on the level of point addition operation. Point addition operation requires the execution of many Montgomery multiplications which consume time. Several techniques are proposed to parallelizing Montgomery multiplications [14, 17]. In this work, we will not concentrate on performing this case. In future work, we will investigate using the proposed architecture to execute Montgomery multiplication to solve the problem of load imbalance.

5 Conclusions

Security services are generally classified into six components: confidentiality, data integrity, authentication, authorization, non-repudiation, and accountability. Cryptography is one of the most popular techniques that can be used to provide sufficient security to the sensitive data. To achieve those issues different protocols have been proposed, all those protocols are based on complicated mathematical operations which are time consuming. Parallel systems, are the most favorable architectures to increase the computing power and expedite these operations. Parallel processing continues to hold the promise of the solution of more complex problems, by connecting a number of powerful processors together into a single system. These connected processors cooperate to solve a single problem that exceeds the ability of stand alone processor. In this work, we propose a generic model to execute any encryption algorithm (symmetric or public) through a parallel-pipelined design.

We address the problem of expediting the public key cryptographic algorithms by using parallel-pipelined design. Therefore, the total computation time is reduced. Parallel/pipelined technique reduces the computation time required to execute the modular multiplication operation, compared to its corresponding values of sequential execution in order to achieve high performance and throughput in public key cryptography. The proposed design makes use of 'M' processing elements to execute different encryption/decryption phases in parallel. Furthermore, it implements the parallelization mechanism on the arithmetic operation level (load-balancing level), where complex arithmetic operations could be divided into small simple arithmetic operations that are executed in parallel. Simulation experiments show that parallel implementations of the aggregated signcryption protocol (as a case study) outperforms the sequential performance (for different values of 'K') for both distinct messages with multiple signers and distinct messages with ordered signatures cases. For the first case the average values of improvement are 50%, 66.6%, 75.5%, 79.9% and 85.4% for M= 2, 3, 4, 6 and 8 respectively. While for distinct messages with ordered signatures case the average values of improvement are 47.5%, 60.6%, 67.3%, 71.5% and 80.4% for M= 2, 3, 4, 6 and 8 respectively.

Although, many researchers have done work in order to speed up the performance of cryptosystems using parallel computing, these algorithms are protocol specific. In contrast, we propose a generic model to execute any encryption algorithm through a parallel-pipelined design.

References

[1] J. Bajard and I. Laurent, "A full RNS implementation of RSA," *IEEE Transactions on Computers*, vol. 53, no. 6, pp. 769-774, 2004.

[2] P. Barret, "Implementing the Rivest Shamir and Adleman public key encryption algorithm on standard digital signal processor," in *Advances in Cryptolgy (Crypto'86)*, LNCS 263, pp. 311-323, Springer-Verlag, 1987.

[3] S. Basu, "A new parallel window-based implementation of the Elliptic Curve point multiplication in multi-core architectures," *International Journal of Network Security*, vol.14, no.2, pp.101-108, Mar. 2012.

[4] D. Boneh, D. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proceedings of Advances in Cryptog-raphy (EUROCRYPT'03)*, LNCS 2656, pp. 416-432, Springer-Verlag, 2003.

[5] W. L. Chang, M. Guo, and M. S. Ho, "Fast parallel molecular algorithms for DNA-based computation: factoring integers," *IEEE Transactions on Nano Bioscience*, vol. 4, no. 2, pp. 149-163, 2005.

[6] C. C. Chen, H. Chien, and G. Horng, "Cryptanalysis of a compact certificateless aggregate signature scheme," *International Journal of Network Security*, vol.18, no.4, pp.793-797, July 2016

[7] M. Ciet, N. Michael, P. Eric, and J. J. Quisquater, "Parallel FPGA implementation of RSA with residue number systems-can side-channel threats be avoided?," in *Proceedings of 46th Midwest IEEE Symposium on Circuits and Systems*, vol. 2, pp. 806-810, 2003.

[8] M. Damrudi and N. Ithnin, "Parallel RSA encryption based on tree architecture," *Journal of the Chinese Institute of Engineers*, vol. 36, no. 5, pp. 658-666, 2012.

[9] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

[10] V. Digraskar, A. Chirde, A. Jagtap, and A. Deasi, "Parallel computation of advance encryption standard algorithm for performance improvement," in *Proceedings of International Conference on Recent Trends in Engineering Science and Technology (ICRTEST'17)*, pp. 139–142, Jan. 2017.

[11] V. Digraskar, A. Chirde, A. Jagtap, and A. Deasi, "Secure file transmission using parallel Aes algorithm," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no.3, Mar. 2017.

[12] C. L. Duta, G. Michiu, S. Stoica, and L. Gheorghe, "Accelerating encryption algorithms using parallelism," in *Proceedings of 19th International Conference on Control Systems and Computer Science*, May 2013.

[13] G. F. El Kabbany, H. K. Aslan, and M. N. Rasslan, "A design of a fast parallel-pipelined implementation of AES: Advanced Encryption Standard," *International Journal of Computer Science & Information Technology*, vol. 6, no. 6, pp: 39-45, Dec. 2014.

[14] J. Fan, K. Sakiyama, and I. Verbauwhede, "Elliptic curve cryptography on embedded multicore systems," *Design Automation for Embedded Systems*, vol. 12, no. 3, pp. 231–242, 2008.

[15] C. Gentry and Z. Ramzan, "Identity based aggregate signature," in *Proceedings of International Workshop on Public Key Cryptography (PKC'06)*, pp 257-273, 2006.

[16] J. GroBschadl, "High-speed RSA hardware based on Barret's modular reduction method," in *International Workshop on Cryptographic Hardware and Embedded Systems (CHES'00)*, LNCS 1965, pp. 191–203, Springer-Verlag, 2000.

[17] N. Guillermin, "A high speed coprocessor for elliptic curve scalar multiplications over FP," in *Proceedings of 12th International Conference on Cryptographic Hardware and Embedded Systems (CHES'10)*, pp. 48-64, 2010.

[18] M. Hossain, E. Saeedi, and Y. Kong , "Point-multiplication architecture using combined group operations for high-speed cryptographic applications," *PLoSOne*, vol. 12, no. 2, May 2017.

[19] K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory*, 2ed., Springer-Verlag, 1990.

[20] D. Knuth, *Semi Numerical Algorithms*, The Art of Computer Programming, vol. 2, Addison-Wesley, Reading, MA, USA, 1969.

[21] Y. Lin, C. Lin, and D. Lou, "Efficient parallel RSA decryption algorithm for many-core GPUs with CUDA," in *Proceedings of International Conference on Telecommunication Systems, Modelling and Analysis (ICTSM'12)*, pp. 85-94, 2012.

[22] Q. Liu, F. Ma, D. Tong, and X. Cheng, "A Regular Parallel RSA Processor," in *Proceedings of 47th Midwest Symposium on Circuits and Systems (MWSCAS'04)*, vol. 3, pp. III-467-70, 2004.

[23] A. Lysyanskaya, S. Micali, L. Reyzin, and H. Shacham, "Sequential aggregate signatures from trapdoor permutations," in *International Conference on the Theory and Applications of Cryptographic Techniques, (EUROCRYPT'04)*, pp. 74-90, 2004.

[24] S. Mahajan and M. Singh, "Analysis of RSA algorithm using GPU programming," *International Journal of Network Security & Its Applications*, vol. 6, no.4, 2014.

[25] A. Menezes, S. Vanstone, and P. Oorschot, *Handbook of Applied Cryptography*, CRC Press, Inc., Boca Raton, FL, USA, 1996.

[26] M. Rasslan and H. Aslan, "On the security of two improved authenticated encryption schemes," *International Journal of Security and Networks*, vol. 8, no. 4, pp. 194-199, 2013.

[27] A. Rawat and S. Walfish, *A Parallel Signcryption Standard Using RSA with PSEP*, Technical Report, 2003.

[28] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[29] V. H. Sathawane and T. Diwan, "An optimization parallel computation of advance encryption algorithm using Open-MP- a review," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 7, no. 2, pp. 384-386, Feb. 2016.

[30] L. Savu, "Signcryption scheme based on Schnorr digital signature," *Journal of Software Engineering and Applications*, vol. 05, no. 02, Jan. 2012.

[31] S. Saxena and B. Kapoor, "State of the art parallel approaches for RSA public key based cryptosystems," *International Journal on Computational Sciences & Applications*, vol.5, no.1, Feb. 2015.

[32] C. P. Schnorr, "Efficient identification and signatures for smart cards," in *Advances in Cryptology (Crypto'89)*, LNCS 435, pp. 239-252, Springer-Verlag, 1990.

[33] S. Tang, K. Tsui, and P. Leong, "Modular exponentiation using parallel multipliers," in *Proceedings of IEEE International Conference on Field-Programmable Technology*, pp. 52-59, Dec. 2003.

[34] C. Wu, D. Lou, J. Lai, and T. Chang, "Fast parallel exponentiation algorithm for RSA public-key cryptosystem," *Informatica*, vol. 17, no. 3, pp. 445-462, 2006.

[35] D. Zhang, H. Hang, and X. Bi, "Comparison and analysis of GPGPU and parallel computing on multicore CPU," *International Journal of Information and Education Technology*, vol. 2, no. 2, pp.185-187, 2012.

[36] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) cost (signature) + cost (encryption)," in *Proceedings of the International Conference of CRYPTO'97*, LNCS 1294, pp. 165-179, Springer-Verlag, 1997.

Biography

Mohamed Rasslan is an Assistant Professor at Electronics Research Institute, Cairo, Egypt. He received the B.Sc., M.Sc., degrees from Cairo University and Ain Shams University, Cairo, Egypt, in 1999 and 2006 respectively, and his Ph.D. from Concordia University, Canada 2010. His research interests include Cryptology, Digital Forensics, and Networks Security.

Ghada El-kabbany is an Associate Professor at Electronics Research Institute, Cairo- Egypt. She received her B.Sc. degree, M.Sc. degree and Ph.D. degree in Electronics and Communications Engineering from the Faculty of Engineering, Cairo University, Egypt. Her research interests include High Performance Computing (HPC), Image Processing, Computer Network Security and Digital Forensics.

Heba Kamal Aslan is a Professor at Electronics Research Institute, Cairo-Egypt. She received her B. Sc. degree, M. Sc. degree, and Ph. D. degree from Faculty of Engineering, Cairo University, Egypt in 1990, 1994 and 1998 respectively. Her research interests include Key Distribution Protocols, Authentication Protocols, Logical Analysis of Protocols and Intrusion Detection Systems.

Detection of Impersonation Attack in MANET Using Polynomial Reduction Algorithm

Perumal Raghavan Kavitha, Rajeswari Mukesh
(Corresponding author: P. Kavitha)

Department of Computer Science and Engineering, Hindustan Institute of Technology and Science
Padur, Chennai, Tamil Nadu, India
(Email: kavithashree2007@gmail.com)

(Received Mar. 15, 2017; revised and accepted June 25, 2017)

Abstract

Secure routing among multiple routers become a complex process in Mobile Ad-hoc NETworks (MANETs) due to the presence of adversary nodes. An adversary node which compromises the honest node as well as it will use all secrets of compromised node. By these capabilities, the attacker launches impersonation attacks against secure transmission. Strengthening the personal key and monitoring the behavior of each node is crucial avoid impersonation attack. This paper proposed the Enhancing Polynomial reduction algorithm for Secure routing against Impersonation in Clustered MANET to tackle above secure routing issues. To provide the personal key with less key storage overhead, the clusters network based on the flexible weight clustering algorithm. To avert impersonation, the polynomial reduction algorithm which initially exploits the strong key authentication via two separate transactions which forward the set of primitive polynomial with key function and the same with anyone random number in a successive manner. Also, the polynomial reduction algorithm considers the acknowledgments with attached hop-counts from the intermediate routers to identify the impersonation when the attacker acts as an intermediate router resulting in high control overhead. Therefore, the proposed introduces the Energy Efficient Polynomial Reduction (EPR) algorithm that reduces the overhead by sending the consolidated acknowledgment for same cluster nodes.

Keywords: Clustering; Impersonation Attack; MANET; Polynomial Reduction Algorithm

1 Introduction

A collection of mobile nodes forms a self-configuring and temporary network using wireless links, named as Mobile Ad-hoc NETwork (MANET). The highly dynamic nature of MANET suffers the direct wireless communication due to the limited bandwidth and transmission range. For using multi-hop routing strategies, it is necessary to establish the wireless communication among remote mobile nodes [18]. Due to multiple intermediate routers in multihop routing, a secure packet transmission is not assured in the presence of impersonation attacks. In that attack compromises the legitimate node to launch some attacks against secure routing. The key authentication technique ensures a secure forwarding among multiple intermediate routers. The personal key authentication technique supports to avoid the impersonation. A private key is hacked by overhearing then node is stored, which leads to the possibility of launching impersonation attack [3, 12]. Furthermore, the key storage overhead and personal key maintenance escalate in each node in the group. Therefore, to ensure secure routing, it is necessary to consider the impact of impersonation with considerable overhead.

The standard works estimate the impersonation by employing a polynomial reduction method [7, 16]. The polynomial reduction algorithm identifies the impersonation according to the primitive and checks polynomials. However, the wireless medium gives the possibilities to overhear the key authentication. Thus, the attacker may trace the key through overhearing and initiate the false routing. Moreover, also this algorithm verifies the intermediate routers in the path by receiving the acknowledgments. Thus, it makes the high control overhead and network traffic. To tackle these issues, the Providing Enhanced Polynomial reduction algorithm for Secure routing against Impersonation in Clustered MANET is proposed to augment the key authentication methodology and identify the impersonation using Energy efficient Polynomial Reduction (EPR) algorithm with reduced routing overhead.

The significant strategies which contributions of the paper are as follows:

- The proposed protocol assures secure and reliable routing in the highly dynamic network by detecting and isolating the impersonation attacker in the routing path using EPR algorithm.

- By partitioning the network into several clusters using a weight clustering algorithm, and maintaining the cluster group key and separate personal key for authentication, restricts the nodes to initiate the malicious routing attacks with considerable key storage overhead.

- By forwarding the primitive polynomial set with key function and the same random number via two separate transaction steps in a successive manner to improve authentication.

- By generating the consolidated acknowledgment for the same nodes using group ID verification of previous and next-hop nodes, the protocol significantly lowers the acknowledgment overhead as well as escalates the energy storage.

Next section focuses exclusive study on MANET security routing protocols, Section 3 represents the problem statement, Section 4 shows System and Attack Model in MANET; Section 5 gives the overview of Proposed EPR (PEPR) protocol; Section 6 gives detailed performance evaluation for PEPR. and final section concludes the paper.

2 Related Works

Even though the routing protocol establishes the multihop routing using multiple intermediate routers, the secure and reliable data delivery is unlikely achieved if any impersonation attacker is selected as an intermediate router. The impersonation attacks target the data transmission functionality between source and destination. The adversary nodes compromise the genuine node to simply launch impersonation, dropping, and modification of the original data content to reduce the network performance. Several secure routing approaches have been proposed against such attackers in MANET [6, 8, 24].

The asymmetric group key provides the security with the public and private key. In this, the digital signature and data encryption are performed using the public key. The individual decryption key is maintained as confidential [25, 27]. Furthermore, the dropping attackers drop the received packets, either for saving their resources or for deliberately disrupting regular communications [9]. Observing the routing behavior of nodes is important to combat with a different variety of dropping attacks such as black-hole, gray-hole, and wormhole attacks [14, 20]. Such attacks drop the packets entirely or selectively. This kind of attackers can be identified by continuously monitoring the behavior of nodes during data forwarding. For example, the sender node can trace the data forwarding records of a particular packet at each node by formulating a Renyi-Ulam game based tracing. In this, the first hop data packet forwarded is no longer considered as a misbehaving node [17]. The detection of packet dropping is accounted in the presence of link error and collision using the correlation between them. The correlation enhances the accuracy of drop detection with Homomorphic Linear Authenticator (HLA) based public auditing architecture. The HLA ensures the effectiveness of a packet loss report that is delivered by individual nodes [21]. Similarly, counting the number of lost packets caused by link error can assist to detect malicious packet dropper [13, 26]. Moreover, the clustered network handles the routing issues on routing and the cluster partitions efficiently identify the malicious node. The highly efficient node is selected as a CH for providing the secure communication via cooperative nodes [5, 19].

To support the secure routing, it is essential to consider both the dropping and modification attacks to maintain reliability and confidentiality without modification using cryptography based encryption method [11]. Several cryptography based routing protocols have been proposed for secure data forwarding [6]. The Advanced Encryption Standard (AES)-based routing algorithm helps to consolidate the data integrity using hash codes containing the data packets. The AES employed with Triple Data Encryption Standard (TDES) assist in preventing the Packet Dropping and Message Tampering Attacks [23]. The polynomial scheme supports the detection of attacker nodes. In [16], the clustering provided additional security and secured key transmission between Cluster Head (CH) and members. The soft computing assists in detecting the irregularity and misuse. In [22], the same group based key transmission is exploited to assure the secure routing. The Logical Key Hierarchy (LKH) is employed to accomplish the personal key refreshment. The Hop Count Based Key Selection (HBKS) technique uses symmetric key based authentication and key pre-distribution method to find the impersonation, dropping, and modification attackers. This method can only identify the attacker who is acting as an intermediate router, and it is not supported to prevent the attacker who serves as a source or destination. When the modification attacker performs quite brilliantly, identification of modification attack becomes more difficult due to the possibility of changes in the original message [1]. The asymmetric encryption reduces the key distribution burden using a pair of keys rather than a symmetric encryption method [15]. For example, the Optimized Link State Routing Secured by Dynamics key (OLSR-SDK) [10] protocol exploits the asymmetric key distribution. It deploys the dynamic keys and secures the traffic routing without degradation of network performance. The Radio Aware Optimized Link State Routing (RA-OLSR) protocol applies high-level security mechanism based on Identity-Based Encryption (IBE) which eliminates the need of verifying the public key authenticity [4].

3 Problem Statement

The secure routing is mainly depending on the key authentication and the group key as well as personal key maintenance on each node that increases the key storage overhead. The conventional methods exploit the direct key transaction method. The main limitation of the wireless network is that other nodes can overhear the communication between any two nodes. This hole provides the chance for the impersonation attacker to hack the key transaction via overhearing, if it happens, the attackers impersonate the honest nodes which launch the severe malicious attacks on the routing using that traced personal key. The polynomial reduction algorithm aids to recognize the impersonation attack. Even though, it becomes complex when the credential information is hacked. To deal with this, the strong key authentication process is essential. Also, the conventional polynomial reduction methodology escalates the acknowledgment overhead and energy degradation even it can identify the impersonation attack it attacks on the current routing path may involve in the upcoming routing process due to lack of the attacker isolation. Hence, it is essential to forward the packets in a secure and reliable manner using exact impersonation attacker identification and attacker node isolation.

4 System and Attack Model

The network area of MANET is represented as $G(V,E)$ in which V represents a set of mobile nodes and E represents the direct communication links between two mobile nodes V. Transmission range of each node is represented as R. The nodes V can impersonate the legitimate nodes that launch some attacks against secure transmission. In a flexible weight clustering, network, $G|\{C^i, C^{i+1}, C^{i+2}, \cdots, C^{i+n}\}$ and (CH^i, CM^N) C^i; $(N = 1, 2, \cdots)$. The CH^i and CM^N exchange the key via separate two transaction steps to avoid the key transaction overhearing. Thus, the CM^N can initiate the routing without any impact of impersonation. The CH^i authenticates and verifies the CM^N uses a set of primitive and checks polynomials. The set of primitive polynomial contains $f(a,b,c)$, in this, a and b are the generalized components and c is the specific report which is generated based on the generalized components. The $f(a,b,c)$ and $f(a,b,c,R)$ are forwarded in separate successive manner. R is the random number. The PEPR has implemented the EPR algorithm for predicting the impersonation with reasonable overhead. In this, S sends $\{Encrypt(data), X^{h-n}\ (n = 0, \cdots, h)\}$ to its next-hop and the intermediate routers forward it until reaching the D. Each hop-router replies the S by sending the ACK with deducted hop-counts $(X^{h-1}, X^{h-2}, X^{h-3}, \cdots, X^{h-h})$. If more than one node C^i, sends the consolidated ACK to S, S identifies the impact of impersonation from the received ACKs. The identified attackers are isolated from the network performance by calculating RTT values of each router and hypothesis of other nodes. Assume the model that adversary can overhear the key transaction and impersonate the legitimate node to intercept the communication when it hacks the personal key of the nodes.

5 Overview of Proposed EPR Protocol

The most destructive attack in MANET is impersonation attack. The cluster based group key management which reduces the overhead. However, it has the possibility to track the confidential communication between nodes.In this, the confidential message can be decrypted by all the nodes in the same group using the same cluster key. Also, if the individual personal key of any node is tracked by an attacker, it can launch the impersonation attack in routing performance. Instead, two separate transactions based individual personal key sharing is a feasible technique to ensure the secure routing against such attackers in MANET.

The CH is responsible for providing the individual personal keys to all the nodes in the same cluster. In this, the separate personal key is given to the member nodes by CH via successive transactions, instead of directly exchanging the personal key. The incorporated value of both the set of a primitive polynomial and the same set with a random number are forwarded in two transactions that creates the confusion on exact key function overhearing. Thus, it can reduce the possibility of tracking the personal key exchange regarding overhearing. Additionally, in polynomial reduction, the overhead and energy are taken into account for improving the routing performance. When more than one node in the same cluster acts as intermediate routers, the Energy efficient Polynomial Reduction (EPR) algorithm reduces the overhead by sending the single consolidated acknowledgment instead of individual acknowledgment

5.1 Enhancing Security in Clustered MANET

The conventional routing methods propose cryptography based key encryption using public and private keys for reducing impersonation attacks in MANET. In case, the private key of a node is leaked, the impersonation attacker may throw the security issues by compromising the legitimate node. Moreover, the key maintenance of each node escalates the storage overhead. To maintain the trade-off between security and key storage overhead, cluster based key distribution model is exploited. To get away from frequent key refreshment for the entire network, the rekeying process is applied only to clusters. The secure routing depends on the key management, and also depends on the network parameter such as energy. Therefore, the flexible weight clustering approach is proposed to play a vital role in providing the energy efficient, secure routing.

The flexible weight clustering method allows the network partitions into groups of entities called clusters. Each cluster has a Cluster Head (CH) and cluster members who are in one-hop communication from CH. It is elected based on the combination of metrics such as mobility, density, energy, connectivity, distance, and many other metrics. Each node in the cluster observes the routing metrics of its neighbor nodes using broadcast messages. Commonly, the node which has high connectivity and high battery power that has announced as a CH to its neighbors is shown in Equation (1), because such nodes are favorable to the accurate and energy efficient routing in a dynamic network.

In this, each node calculates the cluster score to itself and its neighbors for selecting CH based on connectivity, and battery power. The node which has a high cluster score compared with others announces it as a CH to its neighbors. If a node receives CH announcement from more than one CHs, it chooses any one of the CH depending on its convenience. The node selects one CH with large cluster score, and it joins as a member under the selected CH.

$$N_{(CH)value} = (a \times \frac{N_{\mathrm{Co}}}{N_{\mathrm{Cmax}}}) + (b \times \frac{B_i - B_c}{B_i}). \qquad (1)$$

Where $N_{(CH)value}$ represents the clustering score of each node, N_{Co} and N_{Cmax} represent the original connectivity (number of neighbors) and maximum connectivity of nodes in the network area respectively. B_i is the initial battery power, and B_c is a consumed power. The maximum connectivity is calculatedusing Equation (2). Where, v is the number of nodes in a particular area (HW).

$$\begin{aligned} N_{\mathrm{Cmax}} &= \left[\frac{v}{H \times W}\right] \times R^2 && (2) \\ (a+b) &= 1. && (3) \end{aligned}$$

a and b are the weighting factors of connectivity and energy parameters respectively. It can weigh the parameters to obtain the estimated quantity. The sum of the weighting factors is shown in Equation (3). Algorithm 1 shows the flexible weight clustering algorithm.

Algorithm 1 Flexible Weight Clustering Algorithm

```
Aim: To partition the network into clusters
Input: CH score value of each node
Output: Election of CH and members
Begin
Each node do {
Calculates its own connectivity based on the number of neighbors (Nc);
Calculates the remaining energy level (Ne) based on their initial energy and consumed energy;
Estimates the CH score value according to both Nc and Ne;
Shares their own CH score value to their neighbors;
Compares their own value with other's value;
}
A node $N_i$ that has the high CH score than others broadcasts the CH announcement;
The other nodes that has CH score<Score of $N_i$ forward the reply to $N_i$ to join as a member under $N_i$;
If (Received CH announcement == 1)
    { Node joins under that CH node; }
else (Received CH announcement > 1)
    {Node selects any one CH announcement based on its convenience;
    Node forward the reply to join as a member under the preferred CH;
    }
The Selected CH starts the authentication key sharing process between its members;
End
```

Moreover, CH is appointed in a rotating manner, which means the same node does not elect as a CH continuously. The CH role rotation among real nodes balances the energy consumption among them. The nodes in the cluster cannot involve in any routing performance without having the acknowledgment from its CH. The CH supports the cluster and personal key refreshment and thus, it can avoid the member nodes to launch the impersonation attack using multiple transactions based key authentication in most of the cases.

6 Avoiding Impersonation Using Energy Efficient Polynomial Reduction Algorithm

The polynomial reduction algorithms identify the impact of impersonation in the routing path. In the clustered network, the CH is responsible for distributing the separate personal key to their members. In this, all nodes in the cluster are enabled to maintain the set of a primitive polynomial which contains two generalized components and one specific report. This can be generated by using other two generalized components, which is observable by everyone due to the nature of the wireless medium. By this hole, an attacker can overhear and generate the same specific report as the similar report of the honest node using that generalized component. Thus, the CH mistakenly authorizes the attacker as a legitimate node and provide the key to an attacker resulting in impersonation attack launch. To strengthen the key authentication, the polynomial reduction algorithm initially exploits the two steps in the authentication process with the help of any random number support.

6.1 Step 1: Strengthening the Key Authentication

The polynomial reduction algorithm initializes all mobile nodes and the network with $f(a, b, c)$. In this, a, b, c represents the node ID, all forwarding nodes ID, and all measurement reports respectively. Function a and b are the generalized components and c is the specific report which is based on a and b. The c report is encrypted using the cluster key and stored in that node. The CH knows the node ID as well as the primitive polynomial set of all members. The authentication polynomial is computed for cluster 'Cx' as follows:

$$Auth_x^p(b, c) = \alpha f_x(p, b, c).$$

Where, $f(nodeID, b, c)$ represents the set of primitive polynomial. $\{2, 2^2, 2^3, 2^4, 2^5\}$. Note that the algorithm randomly chooses the value α while computing the authentication polynomial. The algorithm only knows the value of α. Otherwise, no other persons can extrapolate the authentication polynomial of cluster 'Cx'. The purpose of α is to escalate the flexibility of this scheme on the number of compromised nodes. After the computation, the $Auth_x^p(b, c)$ is stored in node 'p'. Subsequently, the algorithm computes the verification polynomial as follows. For each cluster, (x, y).

$$Verf_y^p(a, c) = \beta f_y(a, p, c).$$

Where, $Verf_y^p(a, c)$ is the verification polynomial of cluster 'Cy' and it is stored in node 'p'. In this, $\{2^5, 2^6, 2^7, 2^8\}$, it performs the same role as, and in such 2^t, t is the positive integer. Both the value of α and β is taken into verification using check polynomial β/α. The value of β/α only belongs to the $\{2^0, 2^1, 2^2, 2^3, 2^{4,}2^5, 2^6, 2^7\}$. For example, $2^5/2^5 = 2^0 \cdots 2^8/2 = 2^7$. However, the hole in the polynomial reduction is the report may be generated as equal to the legitimate node by an attacker using the generalized components that lead to the mistake verification. To overcome this, the proposed method exploits the two-step authentication. In this, the incorporated function value of a, b, c and ab, c, rare forwarded in the separate and successive manner via two transactions. R is the random number selected by the node. The both are forwarded in an unpredictable manner that means the $f(a, b, c)$ may send either firstly or secondly as the same the $f(a, b, c, R)$ may send in either firstly or secondly. The random number creates the confusion on the attacker decision about which is the original report. It strongly assists the polynomial reduction algorithm to correctly verify and distribute the key to the legitimate node without the impact of any impersonation. Algorithm 2 shows the separate function transaction algorithm via two steps.

Algorithm 2 Separate Function Transaction Algorithm

```
Aim: To restrict the impersonation as original
Input: The set of primitive polynomial and the random number
Output: Original report identification and verification
Begin
CH Sends the request to $CM^N$ in cluster C ($N = 1, 2, \cdots, n$);
$CM^N$ select the random number R;
$CM^N$ incorporate the random number with $f(a, b, c)$ and generate another $f(a, b, c, R)$;
$CM^N$ forward the two separate functions in two forwarding steps to the CH;
CH receives one function and wait for some time to receive the another function value;
After receiving the both $f(a, b, c)$ and $f(a, b, c, R)$, the CH verifies the $CM^N$;
CH splits the report 'c' and 'R' from both received values;
CH generate the original report based on its own primitive polynomial set for that $CM^N$;
CH compares the report computed by itself with the node report using check polynomials;
If (Report == same)
    { The CH generates the key for that corresponding $CM^N$; }
else
    { The CH identifies the attacker and discards the report; }
End
```

In the above algorithm, the CH authenticates and verifies the node only after receiving the two function values from the same node to avoid the impersonation. Now, both the node and the CH know the report 'c' and creates the key. The node exploits the generated key to encrypt the data during forwarding.

6.2 Step 2: Energy Efficient Polynomial Reduction Algorithm

Strengthening the Key Authentication ensures the nodes to initiate the secure forwarding without any impact of impersonation. In case, the routers misbehave in the discovered shortest routing path based on AODV, which severely makes the routing performance. To overcome these issues, the polynomial reduction algorithm efficiently finds the router impersonation using received acknowledgments with reduced hop-counts (X^h, X^{h-1}, X^{h-2}, X^{h-3}, $\cdots$, X^{h-h}). Every intermediate node sends back the acknowledgment to the sender node that ensures the successful data forwarding in hop-by-hop. However, when the number of acknowledgment escalates on the forwarded data packets, it results in high energy consumption. To tackle this inconvenience, in addition to the security, the proposed method considers both the acknowledgment overhead and energy that is named as the Energy efficient Polynomial Reduction (EPR) algorithm.

The possibility of presenting more than one router in the same cluster is high due to the distributed structure of the MANET. The high possibility reduces the chances of generating the individual acknowledgments in the same cluster. Thus, the consolidated acknowledgment escalates the advantages of improving both the overhead and energy consumption. For example, the representation of EPR algorithm is shown Figure 1.

In Figure 1, S forwards the data packets and estimated remaining hop-count X^4 to the intermediate Router 1 in C2. The Router 1 checks its group list, and it finds the position of Router 2 which is in other cluster C3. The Router 1 in C2 sends the acknowledgment ($X^{4-1} = X^3$) to S (source) after forwarding the data packets to Router 2.

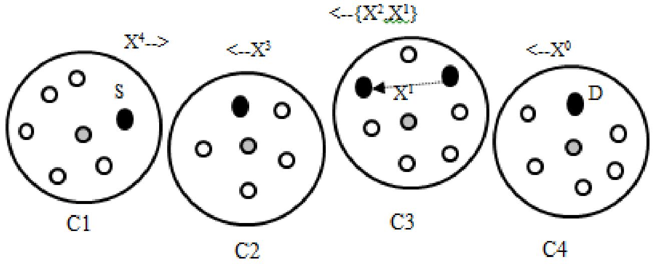


Figure 1: Representation of EPR algorithm

Router 2 in C3 forwards the packet to router3 and the Router 2 identifies the position of router3 by verifying group list. Therefore, it waits for receiving the acknowledgment from router3. After receiving X^1 from router3, the Router 2 creates the consolidated acknowledgment using its $X^{4-2} = X^2$ and $X^{4-3} = X^1$. The consolidated $\{X^2, X^1\}$ is forwarded to S by the Router 2. Finally, the corresponding D sends its acknowledgment $X^{4-4} = X^0$ to S.

The sender node estimates the number of acknowledgments on the number of data packets sent and hop-count between itself and destination that means each hop router forwards one acknowledgment for each data packet. However, due to the acknowledgment consolidation of the proposed method, the number of received acknowledgments may differ from the estimated one. The number of unreceived acknowledgments is predicted using Equation (4).

$$ACK_{(unreceived)} = (P_i \times H) - \sum_{R_{n=1}}^{R_{n=D}} (ACK_{(sent)}). \qquad (4)$$

Where, P_i represents the number of packets and H represents the expected hop-count between sender and destination. Rn indicates the intermediate routers and $(ACK_{(sent)})$ represents the number of acknowledgments sent by each intermediate router. In addition to the destination, every intermediate node is enabled to send an acknowledgment to the sender after transmitting the received packets. Even when the intermediate node increases the proposed methodology can identify the attackers successfully. The unreceived acknowledgments may be affected by impersonation attack, or it may be consolidated with other nodes acknowledgment. There is a need to differentiate from each other to exactly calculate the presence of malicious impersonation attackers using Equation (5).

$$Imp^H = ACK_{(unreceived)} - N^H_{(ACK(cons))}. \qquad (5)$$

Where $N^H_{(ACK(cons))}$ represents the number of hops included in consolidated acknowledgment. The Imp^H implies the presence of impersonation attackers in a path. Algorithm 3 shows the proposed EPR algorithm.

Algorithm 3 The Proposed EPR Algorithm

```
Aim: To successfully predict the impersonation attacker in the routing path
Input: Key authentication and ACKs
Output: Successfully avoid the impersonation
Begin
Step 1. $Auth^p_x(b,c) = f_x(p,b,c)$. Authentication polynomial of cluster $Cx$ for $p$;
Step 2. $Verf^p_y(a,c) = f_y(a,p,c)$. Computes and stores the check polynomial in node $p$; $K_{c_x}$ is the cluster key of $C_x$ stored in node p1;For each cluster $x$ is not equal to $y$;
Step 3. $CH_x \to p : (p|CH_x)$. CH sends the local ID assignment request to each node in $CH_x$
Step 4. $P \to CH_x$: $(f(a,b,c),\ f(a,b,c,R))$. $c$ is the specific report based on the generalized components $(a,b)$ that is encrypted by $K_{c_x}$ and R is the random number selected by node $p$.
Step 5. $c = H((Ea,b)K_{C_x})$. $CH_x$ receives two function values; Calculates the specific report 'c' using its set of primitive polynomial for node $p$.
Step 6. $CH_x$ verifies the report $c$ using check polynomials and get the key.
Step 7. $c$ of $CH_x$ = $c$ of $p$. The report that is generated by cluster head and node is same, the $CH_x$ authorizes the node $p$; The node $p$ encrypts the data using $k_{c_x}$.
Step 8. The node $p$ identify the impersonation among intermediate routers using received ACK with attached hop-counts.
I. $p \to R_n$: $[E(Mesg), X^h)]$. The source node sends the data packet that contains encrypted message by its personal key and expected hop-count $(X^h)$ to reach the destination.
II. $R_n \to R_n + 1$: $[(E(Mesg), X^{h-1}]$.
III. $R_n$ checks the group list to know the current position of the next-hop router $(R_n + 1)$;
IV. If $(R_n + 1C_i)$, $R_n \to p : [X^{h-1}]$. C represents same cluster. Next-hop router is not located in C, $R_n$ forwards the ACK to $p$.
V. If $(R_n + 1C)$,
Begin
$R_n$ wait for some time to receive the acknowledgment from $R_n + 1$;
$R_n + 1$ checks the group list to identify the ID of its previous-hop (Rn) whether it is in the same group ID or not. If yes, follows Step 6;
VI. $R_n + 1 \to R_n : [X^{h-2}]$. After some time (viz $R_n{+}1 \to R_n{+}2 : [(E(Mesg), X^{h-2})])R_n{+}1$ forwards acknowledgment to $R_n$.
VII. $R_n \to p : [\{X^{h-1}, X^{h-2}\}]$. $R_n$ forwards the consolidated acknowledgment to $p$;
The intermediate routers follow the above steps from II to VII until reach the destination (D) that means $D \to p : [X^{h-h}]$. }
End
```

7 Performance Evaluation

The extensive NS2 simulation is applied to validate the performance of proposed EPR Algorithm (PEPR) with the existing Polynomial Reduction Algorithm (ExPRA) [2]. The simulation takes 30, 100 and 200 mobile nodes over an area of 500m x 500m, 1000m x 1000m, 1500m x 1500m. The transmission range of each mobile node is 250m. The overall simulation time is taken of 60 seconds. The proposed method ensures the identification and detection of impersonation attacker with less overhead. The two-step exchange process of primitive polynomial set assists to generate the confusion on attacker overhearing. This process enables the member nodes to share their personal keys with CH in a secret manner. The acknowledgment consolidation for same cluster nodes leads to attaining low overhead. The proposed proves the performance through the detection accuracy, when compared to ExPRA with varying number of attackers.

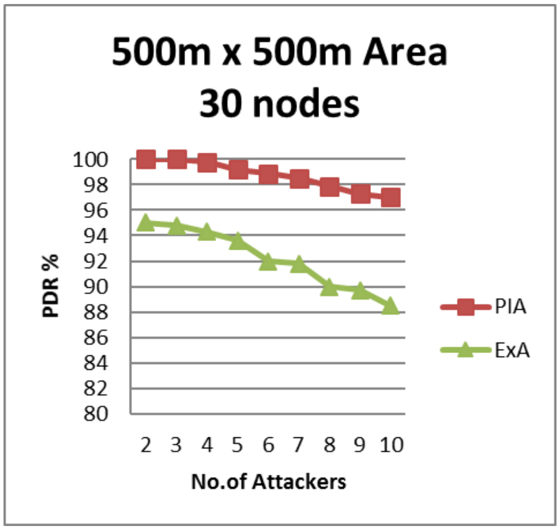


Figure 2: Detection accuracy vs number of attackers for 30 nodes

Figures 2, 3, 4 exposes the attacker detection accuracy of both the proposed EPR and ExPRA with varying number of attackers. Initially, the PEPR has 100 % detection accuracy compared to ExPRA in the presence of 4 attackers. When increasing the number of attackers to 12, the detection accuracy of both the methods gets reduced. The PEPR avoids the attacker impersonation at the sender side using two-step authentication process, and also it exploits the random number with an original primitive polynomial function to confuse the attacker during overhearing. If any impersonation attackers act as an honest one, the CH easily identifies such attackers using check and verification polynomials. Also, if any intermediate node tries to serve as a destination for getting the data, the acknowledgments help to detect the attacker in the path. Then ExPRA detects only 25% attackers at the beginning due to the possibility for overhearing the key authentication. In the presence of 12 attackers, the ExPRA has 85% less detection accuracy than PEPR method. Since, when more than one intermediate routers have the same cluster ID, the attackers have the possibility to create the fake acknowledgment. In such cases, the detection accuracy is lesswhen increasing the number of attackers. However, the PEPR achieves relatively much better detection accuracy than ExPRA method.

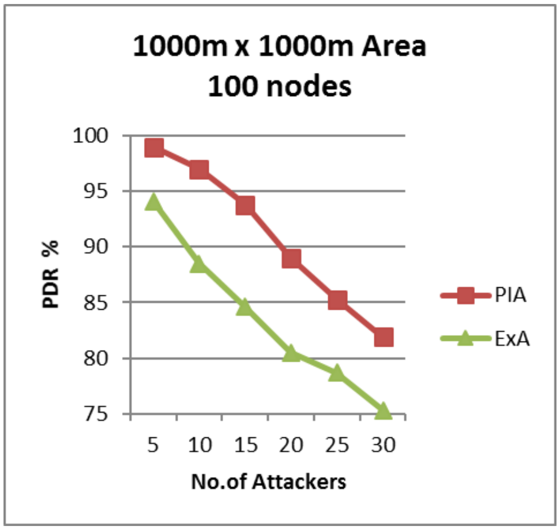


Figure 3: Detection accuracy vs number of attackers for 100 nodes

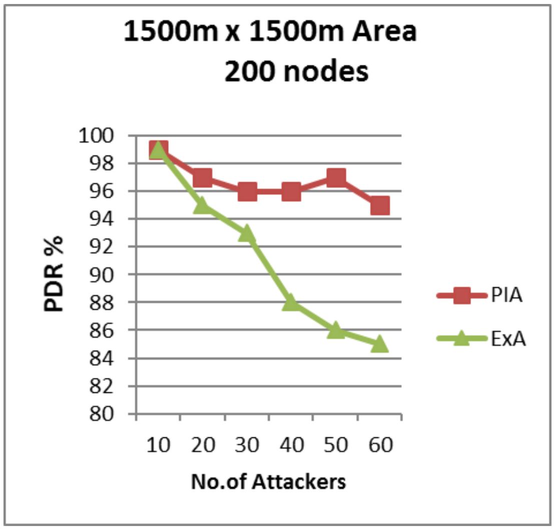


Figure 4: Detection accuracy vs number of attackers for 200 nodes

8 Conclusion

This proposed paper has achieving secure and reliable data forwarding in highly dynamic networks using the detection and elimination of impersonation attackers. By enforcing these two separate transactions of the primitive polynomial set with key function and the same with anyone random number in a Successive manner, the PEPR efficiently restricts the overhearing, as well as an attacker, does not hack the personal of a legitimate node. Thus, a severe impact of impersonation attacker has been reduced in clustered network. The novel clustering algorithm based network partition assist to elect the energy efficient and high worthy CH as well as reduce key overhead. By introducing the Energy efficient Polynomial Reduction (EPR) algorithm, the PEPR detects the attackers in the routing path with less overhead using consolidation method. Also, by exploiting the attackers in the routing path, accurately classifies and eliminates the attackers and announces such attackers to neighbor CHs for preventing the impersonation in the future. Finally, the simulation shows the better performance of PEPR than the existing ExPRA in terms, detection accuracy.

References

[1] K. V. Arya, S. S. Rajput, "Securing AODV routing protocol in MANET using NMAC with HBKS technique", in *IEEE International Conference on Signal Processing and Integrated Networks (SPIN'14)*, pp. 281–285, 2014.

[2] S. Balasubramani, S. K. Rani, S. Rajeswari, "Review on security attacks and mechanism in VANET and MANET", in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp. 655–666, 2016.

[3] M. Barbeau, J. Hall, and E. Kranakis, "Detecting impersonation attacks in future wireless and mobile networks", in *Secure Mobile Ad-hoc Networks and Sensors*, pp. 80–95, 2006.

[4] J. Ben-Othman and Y. I. S. Benitez, "A new method to secure RA-OLSR using IBE", in *Global Communications Conference (GLOBECOM'12)*, pp. 354–358, 2012.

[5] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks", *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[6] J. Chen, J. Wu, "A survey on cryptography applied to secure mobile ad hoc networks and wireless sensor networks", in *Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice*, pp. 262–289, 2010.

[7] J. H. Cho, R. Chen, "Model-based evaluation of distributed intrusion detection protocols for mobile group communication systems", *Wireless Personal Communications*, vol. 60, no. 4, pp. 725–750, 2011.

[8] N. Choudhary, L. Tharani, "A survey of routing attacks in mobile ad hoc network", in *IEEE Security in Wireless Mobile AD Hoc and Sensor Networks*, pp. 85–91, 2007.

[9] S. Djahel, F. Nait-Abdesselam, and Z. Zhang, "Mitigating packet dropping problem in mobile ad hoc networks: proposals and challenges", *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 658–672, 2011.

[10] A. Echchaachoui, A. Choukri, A. Habbani, and M. Elkoutbi, "Asymmetric and dynamic encryption for routing security in MANETs", in *IEEE International Conference on Multimedia Computing and Systems (ICMCS'14)*, pp. 825–830, 2014.

[11] P. D. Gawande, Y. Suryavanshi, "Cryptography based secured advanced on demand routing protocol in MANET's", in *IEEE International Conference on Communications and Signal Processing (ICCSP'15)*, pp. 1478–1481, 2015.

[12] D. Glynos, P. Kotzanikolaou, and C. Douligeris, "Preventing impersonation attacks in MANET with multi-factor authentication", in *IEEE Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'05)*, pp. 59–64, 2005.

[13] T. Hayajneh, P. Krishnamurthy, D. Tipper, and T. Kim, "Detecting malicious packet dropping in the presence of collisions and channel errors in wireless ad hoc networks", in *IEEE International Conference on Communications*, pp. 1–6, 2009.

[14] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "A novel approach for grayhole and blackhole attacks in mobile ad hoc networks", in *Second International Conference on Advanced Computing & Communication Technologies*, pp. 556–560, 2012.

[15] R. K. Kapur, S. K. Khatri, "Secure data transfer in MANET using symmetric and asymmetric cryptography", in *IEEE International Conference on Infocom Technologies and Optimization (ICRITO'15)*, pp. 1–5. 2015.

[16] P. Kavitha, R. Mukesh, "To detect malicious nodes in the mobile ad-hoc networks using soft computing technique", in *IEEE International Conference on Electronics and Communication Systems (ICECS'15)*, pp. 1564–1573, 2015.

[17] W. Kozma Jr, L. Lazos, "Dealing with liars: Misbehavior identification via Renyi-Ulam games", in *International Conference on Security and Privacy in Communication Systems*, pp. 207–227, 2009.

[18] M. K. Marina and S. R. Das, "Routing in mobile ad hoc networks", *Ad Hoc Networks*, pp. 63–90, 2005.

[19] M. M. Morshed, M. R. Islam, "CBSRP: Cluster based secure routing protocol", in *IEEE International Conference on Advance Computing (IACC'13)*, pp. 571–576, 2013.

[20] M. Sadeghi, S. Yahya, "Analysis of wormhole attack on MANETs using different MANET routing protocols", in *Fourth International Conference on Ubiquitous and Future Networks (ICUFN'12)*, pp. 301–305, 2012.

[21] T. Shu, M. Krunz, "Privacy-preserving and truthful detection of packet dropping attacks in wireless ad hoc networks", *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 813–828, 2015.

[22] W. Wang, Y. Wang, "Secure group-based information sharing in mobile ad hoc networks", in *IEEE International Conference on Communications*, pp. 1695–1699, 2008.

[23] I. Woungang, S. K. Dhurandher, V. Koo, and I. Traore, "Comparison of two security protocols for preventing packet dropping and message tampering attacks on AODV-based mobile ad hoc networks", in *IEEE Globecom Workshops*, pp. 1037–1041, 2012.

[24] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in mobile ad hoc networks", *Wireless Network Security*, pp. 103–135, 2007.

[25] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer, "Asymmetric group key agreement", in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 153–170, 2009.

[26] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks", in *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 46–57, 2005.

[27] S. Zhang, Q. Cheng, and X. Wang, "Impersonation attack on two identity-based authenticated key exchange protocols", in *WASE International Conference on Information Engineering*, vol. 2, pp. 113–116, 2010.

Biography

P. Kavitha has acknowledged her B.E and M.E Degree in Computer Science and Engineering from National Engineering College in 2002 and 2004. She is presently pursuing her Ph.D in Hindustan Institute of Technology and Science, Chennai, Tamilnadu, India. She has nearly fourteen years of industrial, academic and research Experience. She is a member of IEEE. Her specialization area is Mobile Ad hoc Networks, Vehicular Ad hoc Networks, Cryptography and Network Security, Data mining and Software Engineering.

Dr. Rajeswari Mukesh has received her Ph.D. degree in Computer Science and Engineering from Jawaharlal Nehru University. Hyderabad. At present, she is a Professor and Head of School of Computing Sciences at Hindustan University, Chennai, Tamilnadu, India. She is guiding 8 PhD candidates. She has published above 10 international journals and attended beyond 15 international and National Conferences. Her area of specialization is Big Data, Biometrics, Adhoc Networks and Cyber Security. She is a Member of IEEE and IET. She has won the Women Engineer award recently from IET.

A New Remote Authentication Scheme for Anonymous Users Using Elliptic Curves Cryptosystem

Xueqin Zhang, Baoping Wang, and Wei Wang
(Corresponding author: Xueqin Zhang)

Software School, Nanyang Normal University
No. 1638, Wolong Road, Wolong District, Nanyang 473000, China
(Email: zhangxueqin01@outlook.com)

(Received Jan. 10, 2017; revised and accepted Apr. 11, 2017)

Abstract

Along with the large-scale proliferation of network and information technology, users can obtain the information resources conveniently via intelligent device. Authentication mechanism is a fundamental tool for ensuring secure communications and the validity of communicating party. In this paper, we propose a new authentication scheme for anonymous users using elliptic curves cryptosystem (ECC) which achieves mutual authentication and forward security. Specifically, we certify the validity of our proposal by employing BAN-logic, which is one of the important formal methods. Further, the performance comparison shows that our scheme is more suitable for application scenarios where efficiency and security concerned.

Keywords: Anonymity; Authentication; BAN-logic; Security

1 Introduction

It is necessary for service providing servers to authenticate remote users in the insecure communication channel when users access resources through networks, remote authentication schemes can provide the convenient and secure mechanism for them to verify each other. However, due to the open nature of the Internet networks, the remote authentication schemes are vulnerable to various attacks. How to design a secure remote authentication scheme to resist these attacks is an important issue for researchers.

In 1981, Lamport [14] firstly proposed a password based remote user authentication scheme using password table. However, the proposed scheme was insecure if the password table was compromised. In password based authentication schemes based on [14], adversaries could easily obtain users' passwords by brute force attack and threat system security, due to the low information entropy of these memorable password. Further, the password table is susceptible to the collision attack.

For solving the inherent limitations of password based authentication schemes, smart card was introduced as the second factor to devise protocols due to the convenience and secure computation. In 2004, Das *et al.* [5] proposed a dynamic ID based remote user authentication scheme using smart cards which allowed users to choose and change their passwords freely without responding to servers. However, later on, some researchers revealed that their scheme was not as much secured as they claimed [2]. In 2009, Wang *et al.* [19] pointed that Das *et al.*'s scheme allowed an attacker to complete the authentication without knowing the password and could not provide the mutual authentication. Meanwhile, the authors proposed a dynamic ID-based authentication scheme and claimed that their scheme was more efficient and secure, as well as keeping the merits of Das *et al.*'s scheme. However, Wang *et al.*'s scheme also could not provide anonymity of users during the authentication and was vulnerable to insider attack, stolen smart card attack.

In the aforementioned authentication schemes, smart card is assumed to be a tamper-resistant device (the parameters stored in smart card cannot be compromised). However, many research results have introduced that sensitive data could be extracted by power analysis and fault injection [12,18]. This is the most critical problem which results in security flaws. And hence, a secure authentication scheme should guarantee security rely on sensitivity of secret parameters, rather than confidence of smart card.

In order to overcome the problems mentioned above, a number of password authentication schemes have been proposed [1, 4, 6, 8–11, 15–17, 20–22]. In 2011, Khan et al. [11] presented a remote authentication scheme using smart card which could provide user anonymity, mutual authentication, session key establishment, and resist various attacks. Later on, He *et al.* [9] showed that Khan *et al.*'s scheme failed to preserve user anonymity. Most recently, Jiang *et al.* [10] illustrated that some previous authentication protocol have a range of ignored security flaws, then they proposed an enhanced authentication scheme and claimed that their scheme could eliminate various malicious attacks. Unfortunately, Wei *et al.* [20] demonstrated that Jiang *et al.*'s scheme could not provide perfect forward secrecy and then presented an improved smart card based authentication scheme.

In this paper, we propose a new remote authentication scheme using elliptic curves cryptosystem (ECC) as well as demonstrating its validity through BAN-logic. BAN logic [3] devotes to validate the beliefs of the involved principals in the protocol and has been widely applied in analyzing the security of authentication schemes. Noticeably, the proposed scheme can achieve various of the required security properties and guarantee high efficiency.

The structure of our paper is organized as follows. In Section 2, we propose a new robust authentication scheme. Subsequently, we prove the completeness of the proposal and analyze its security in Section 3. Then, we evaluate its performance in the next section. At last, Section 5 concludes this paper.

2 Our Proposed Scheme

In this section, we propose a new remote authentication scheme which can preserve user anonymity. Our scheme is divided into four phases: registration phase, login phase, authentication and session key exchange phase and password updating phase. The notations used in our scheme are listed in Table 1.

Table 1: Notations

Notations	Meaning
U_i	The ith user with identity ID_i
S	The remote server
ID_i	The identity of U_i
PW_i	The password of U_i
x	The master secret key of S
a, b	The chosen random numbers
T	The current timestamp
SK	The session key shared among U_i and S
$H(\cdot)$	A one-way hash function
$\oplus$	Exclusive-OR operation
$\|$	String concatenation operation
p, q	Two distinct large primes such that $p = 2q + 1$
$E_p(a, b)$	The elliptic curve over a finite field Z_p
P	P is a generator of order q on the elliptic curve $E_p(a, b)$
Q	S's public key, where $Q = x \cdot P$

2.1 Registration Phase

This phase is invoked whenever U_i initially registers to S. Initially, S chooses two distinct large primes p and q with $p = 2q + 1$. $E_p(a, b)$ is an elliptic curve in the finite field Z_p [7,13]. P is a generator of order q on the elliptic curve $E_p(a, b)$ and q must be large enough, so that the ECDLP is difficult to solve with a polynomial-time algorithm. S uses its master secret key x to compute $Q = x \cdot P \ mod \ p$ and public the parameters $\{P, p, Q, E_p(a, b)\}$.

Step R1. U_i selects his/her identity and password tuple (ID_i, PW_i). Then he/she chooses a random number r to calculate $A_i = H(r \| PW_i)$ and sends $\{ID_i, A_i\}$ to S for registration via a trusted network.

Step R2. After receiving the registration request (ID_i, A_i), S utilizes the master secret key x to compute $X_i = H(ID_i \cdot x \cdot Q)$ and $B_i = X_i \oplus H(ID_i \| A_i)$.

Step R3. Then S writes $\{B_i, p, E_p(a, b), P, Q, H(\cdot)\}$ into the smart card and issues it to U_i through a secure channel.

Step R4. U_i stores r on the received card, which contains $\{r, B_i, p, E_p(a, b), P, Q, H(\cdot)\}$.

2.2 Login Phase

When U_i wants to login S, he/she should insert his/her smart card to the terminal and key ID_i with PW_i, then the smart card performs the following steps (See Figure 1):

Step L1. The smart card generates a random nonce a and computes $A_i = H(r \| PW_i)$, $X_i = B_i \oplus H(ID_i \| A_i)$, $C_i = ID_i \cdot P + a \cdot Q$, $R_i = a \cdot P$, $V_i = X_i \oplus H(a \cdot Q \| T_i)$ where T_i is the fresh current timestamp.

Step L2. After that, U_i transmits the login request $\{C_i, R_i, V_i, T_i\}$ to S by a public channel.

2.3 Authentication and Session Key Exchange Phase

Upon receiving the login request message $\{C_i, R_i, V_i, T_i\}$, U_i and S need to perform the following steps to finish the mutual authentication.

Step A1. S checks the validity of the time stamp T_i, if $T_i' - T_i \leq \triangle T$ holds, S continues to execute the next step.

Step A2. Then S calculates $ID_i \cdot P = C_i - x \cdot R_i$ and $V_i^* = H(x^2 \cdot ID_i \cdot P) \oplus H(x \cdot R_i \| T_i)$. If the computed value V_i^* is equal to the received V_i, S will execute the following steps and if $V_i^* \neq V_i$, the login request will be rejected.

Step A3. After the verification of U_i, S chooses a random number b and current timestamp T_s to compute $R_s = b \cdot P$ and $V_s = H(ID_i \cdot P \| b \cdot R_i \| T_s)$. Then S replies the message $\{R_s, V_s, T_s\}$ to U_i via public channel.

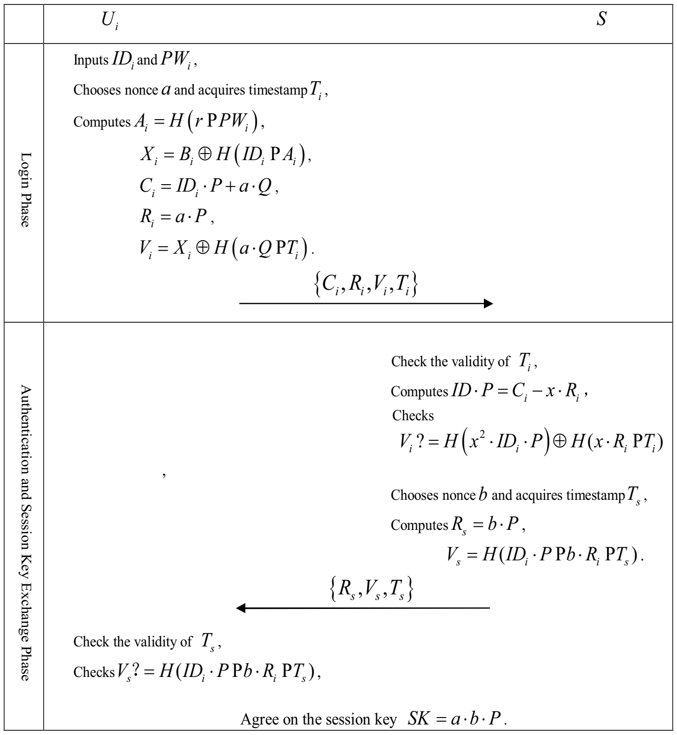

U_i
S
Login Phase
Inputs ID_i and PW_i,
Chooses nonce a and acquires timestamp T_i,
Computes $A_i = H(r \mathrm{P} PW_i)$,
$X_i = B_i \oplus H(ID_i \mathrm{P} A_i)$,
$C_i = ID_i \cdot P + a \cdot Q$,
$R_i = a \cdot P$,
$V_i = X_i \oplus H(a \cdot Q \mathrm{P} T_i)$.
$\{C_i, R_i, V_i, T_i\}$
Authentication and Session Key Exchange Phase
Check the validity of T_i,
Computes $ID \cdot P = C_i - x \cdot R_i$,
Checks
$V_i ? = H(x^2 \cdot ID_i \cdot P) \oplus H(x \cdot R_i \mathrm{P} T_i)$
,
Chooses nonce b and acquires timestamp T_s,
Computes $R_s = b \cdot P$,
$V_s = H(ID_i \cdot P \mathrm{P} b \cdot R_i \mathrm{P} T_s)$.
$\{R_s, V_s, T_s\}$
Check the validity of T_s,
Checks $V_s ? = H(ID_i \cdot P \mathrm{P} b \cdot R_i \mathrm{P} T_s)$,
Agree on the session key $SK = a \cdot b \cdot P$.

Figure 1: Login phase and authentication and session key exchange phase

Step A4. Upon receiving the replication, the smart card verifies the freshness of the time interval between T'_s and T_s, where T'_s is the current time when the mutual authentication message is received. Then the smart card computes $V'_s = H(ID_i \cdot P \| a \cdot R_s \| T_s)$ and checks whether V_s is equal to the computed one. If they are equal, the legitimacy of S is verified by U_i; otherwise, the smart card terminates this session.

After finishing the mutual authentication, U_i and S agree on the common session key $SK = a \cdot b \cdot P$.

2.4 Password Updating Phase

When U_i wants to update his/her password without the help of S, U_i should insert his/her smart card into a card reader and input ID_i and PW_i.

Step P1. U_i is allowed to input a new password PW_i^{new}.

Step P2. The smart card computes $A_i = H(r\|PW_i)$, $A_i^{new} = H(r\|PW_i^{new})$, $B_i^{new} = B_i \oplus H(ID_i\|A_i) \oplus H(ID_i\|A_i^{new})$, and stores B_i^{new} into the smart card to replace B_i and performs an password update successfully.

3 Security Analysis of Our Scheme

In this section, we demonstrate the security performance of our proposed authentication protocol.

3.1 Validity Proof Based on BAN-logic

BAN-logic [3] is a logic belief for analyzing information exchange protocols. Note that, it helps users to ensure whether exchanged messages are trustworthy and secured against eavesdropping. In this section, we demonstrate the completeness of the proposed scheme using BAN-logic.

In the following, we define some notations used in the proof:

$\mathcal{P} \mid\equiv X$: The principal $\mathcal{P}$ believes X.

$\sharp(X)$: The formula X is fresh.

$\mathcal{P} \Rightarrow X$: The principal $\mathcal{P}$ has jurisdiction over X.

$\mathcal{P} \triangleleft X$: The principal $\mathcal{P}$ sees X.

$\mathcal{P} \mid\sim X$: The principal $\mathcal{P}$ once said the statement X.

(X, Y): The formula X or Y is the part of (X, Y).

$\langle X \rangle_Y$: The formula X is combined with Y.

$\{X\}_Y$: This represents the formula X is message and it is encrypted under the key Y.

$\mathcal{P} \overset{k}{\longleftrightarrow} \mathcal{Q}$: The principals $\mathcal{P}$ and $\mathcal{Q}$ communicate with each other with the shared key k. Note that, k will never be known to any other principals.

$\mathcal{P} \overset{k}{\rightleftharpoons} \mathcal{Q}$: $\mathcal{P}$ and $\mathcal{Q}$ shared a secret k, which is possibly known to other principals trusted by them.

SK: the formula SK represents the session key used in the current session.

Let present several logical postulates which is essential for the demonstration as follows:

- The message-meaning rule: $\frac{\mathcal{P}|\equiv\mathcal{Q}\overset{k}{\longleftrightarrow}\mathcal{P},\mathcal{P}\triangleleft\{X\}_k}{\mathcal{P}|\equiv\mathcal{Q}|\sim X}$, $\frac{\mathcal{P}|\equiv\mathcal{Q}\overset{k}{\rightleftharpoons}\mathcal{P},\mathcal{P}\triangleleft\langle X\rangle_k}{\mathcal{P}|\equiv\mathcal{Q}|\sim X}$.
- The freshness-conjuncatenation rule: $\frac{\mathcal{P}|\equiv\sharp(X)}{\mathcal{P}|\equiv\sharp(X,Y)}$.
- The nonce-verification rule: $\frac{\mathcal{P}|\equiv\sharp(X),\mathcal{P}|\equiv\mathcal{Q}|\sim X}{\mathcal{P}|\equiv\mathcal{Q}|\equiv X}$.
- The jurisdiction rule: $\frac{\mathcal{P}|\equiv\mathcal{Q}\Rightarrow X,\mathcal{P}|\equiv\mathcal{Q}|\equiv X}{\mathcal{P}|\equiv X}$, $\frac{\mathcal{P}|\equiv(X,Y)}{\mathcal{P}|\equiv X}$, $\frac{\mathcal{P}\triangleleft(X,Y)}{\mathcal{P}\triangleleft X}$, $\frac{\mathcal{P}|\equiv\mathcal{Q}|\sim(X,Y)}{\mathcal{P}|\equiv\mathcal{Q}|\sim X}$.

Here are some the authentication goals which will be proved for the proper mutual authentication and the agreement of the session key of our scheme.

Goal 1: $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S)$

Goal 2: $S \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S)$.

Next, the corresponding idealised protocol (transmitted by the conventional description of our scheme) is then:

Message 1: $U_i \rightarrow S$: $(C_i, \{R_i, T_i\}_{\langle ID_i \rangle_x}, T_i)$

Message 2: $S \rightarrow U_i$: $(T_s, \langle R_s, T_s \rangle_{\langle ID_i \rangle_{R_i}})$.

The following assumptions are presented to further analyze our scheme:

Assumption 1: $U_i \mid\equiv (U_i \overset{\langle ID_i \rangle_{R_i}}{\rightleftharpoons} S)$

Assumption 2: $S \mid\equiv (S \overset{\langle ID_i \rangle_x}{\longleftrightarrow} U_i)$

Assumption 3: $U_i \mid\equiv \sharp(T_s)$

Assumption 4: $S \mid\equiv \sharp(T_i)$

Assumption 5: $S \mid\equiv U_i \Rightarrow (R_i, T_i)$
Assumption 6: $S \mid\equiv U_i \Rightarrow (R_s, T_s)$
Assumption 7: $U_i \mid\equiv a$
Assumption 8: $S \mid\equiv b$

Based on the aforementioned assumptions and the defined logical postulates, we present the main steps of analysis of our scheme as follows:

When S receiving Message 1, we can prove that:

$$S \triangleleft (C_i, \{R_i, T_i\}_{\langle ID_i \rangle_x}, T_i).$$

Based on the jurisdiction rule, we can prove that:
$S \triangleleft \{R_i, T_i\}_{\langle ID_i \rangle_x}$.

Based on Assumption 2 and the message-meaning rule, we can prove that:
$S \mid\equiv U_i \mid\sim (R_i, T_i)$.

Based on Assumption 4 and the freshness-conjuncatenation rule, we can prove that:
$S \mid\equiv \sharp(R_i, T_i)$.

Based on the proved $S \mid\equiv U_i \mid\sim (R_i, T_i)$ and the nonce-verification rule, we can prove that:
$S \mid\equiv U_i \mid\equiv (R_i, T_i)$.

Based on Assumption 5 and the jurisdiction rule, we can prove that:
$S \mid\equiv (R_i, T_i)$.

Based on the jurisdiction rule, we can prove that:
$S \mid\equiv R_i$.

Based on $SK = H(b \cdot R_i)$ and Assumption 8, we can prove that:
$S \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S)$ (*Goal 2*).

When U_i receiving Message 2, we can prove that:
$U_i \triangleleft (T_s, \langle R_s, T_s \rangle_{\langle ID_i \rangle_{R_i}})$.

Based on the jurisdiction rule, we can prove that:
$U_i \triangleleft \langle R_s, T_s \rangle_{\langle ID_i \rangle_{R_i}}$.

Based on Assumption 1 and the message-meaning rule, we can prove that:
$U_i \mid\equiv S \mid\sim (R_s, T_s)$.

Based on Assumption 3 and the freshness-conjuncatenation rule, we can prove that:
$U_i \mid\equiv \sharp(R_s, T_s)$.

Based on the proved $U_i \mid\equiv S \mid\sim (R_s, T_s)$ and the nonce-verification rule, we can prove that:
$U_i \mid\equiv S \mid\equiv (R_s, T_s)$.

Based on Assumption 6 and the jurisdiction rule, we can prove that:
$U_i \mid\equiv (R_s, T_s)$.

Based on the jurisdiction rule, we can prove that:
$U_i \mid\equiv R_s$.

Based on $SK = a \cdot R_s$ and Assumption 7, we can prove that:
$U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S)$ (*Goal 1*).

3.2 Security Analysis on Possible Attacks

Here, we present that our scheme has the ability to withstand common attacks and achieves some available security properties. Firstly, we assume that the problem list as follows is difficult to solve in polynomial-time. In other words, there are no efficient polynomial-time algorithm to solve it [7]: Given three points $P, s \cdot P, t \cdot P \in E_p(a, b)$, where $s, t \in Z_p^*$, the computation Diffie-Hellman problem(CDHP) is to find the point $(s \cdot t) \cdot P$ on $E_p(a, b)$.

3.2.1 Preserving Anonymity and Nontraceability

The transmitted messages via the public communication channels may leak some useful information about user's identity or activities. In our scheme, the blinded parameter $C_i = ID_i \cdot P + a \cdot Q$ is the only parameter by which server can identify users, and it is session variant due to the random number a. Even if the adversary has obtained the login request C_i, R_i, V_i, T_i of U_i, he/she also cannot compute the $a \cdot Q$ with Q and R_i due to the hardness of computation Diffie-Hellman problem. Further, he/she is unable to retrieve $ID_i \cdot P$ from C_i, while $ID_i \cdot P$ is the only specific static element in the transmitted messages. Hence, our scheme can withstand user anonymity and traceability breach.

3.2.2 Perfect Forward Secrecy

The finished conversation should be confidential even the long-term master secret key compromised. In our scheme, the session key $SK = a \cdot b \cdot P$ is computed with the contribution of a and b. The adversary can only compute SK by two element R_i, R_s which should solve the computation Diffie-Hellman problem. Thus the attacker cannot compute the previously generated session keys and our remote authentication scheme provides the property of perfect forward secrecy.

3.2.3 Off-line Password Guessing Attack

Many existing works suffer from off-line password guessing attack because the low entropy password could be easily obtained by brute force attack. In our proposal, each parameters of the login request are attached with random numbers. Note that, even the attacker has eavesdropped and recorded the login request C_i, R_i, V_i, T_i in the login phase, it is useless for guessing password without knowing $H(a \cdot Q \| T_i)$. Even if the adversary has gotten U_i's smart card and extracted its secret parameters $\{r, B_i, p, E_p(a, b), P, Q, H(\cdot)\}$, he/she cannot construct a equation to verify the correctness of disclosed password. And hence our proposed scheme can resist off-line password guessing attack with smart card security breach.

3.2.4 Forgery Attack

In the scenario of forgery attack, someone masquerades as other legitimate users for illegal access. In our scheme, in order to impersonate as a legal user, the adversary needs to forge a legal login request message C_i, R_i, V_i, T_i. However, he/she cannot compute the C_i and V_i without knowing ID_i and the key X_i generated by S. In a extend scenario, even if the attacker has extracted the secret values $\{r, B_i, p, E_p(a, b), P, Q, H(\cdot)\}$ stored in U_i's smart

Table 2: Comparisons of functionality

Functionality	[15]	[22]	[10]	[20]	Ours
Resistance of forgery attack	No	Yes	Yes	No	Yes
Resistance of off-line password guessing attack	Yes	Yes	Yes	Yes	Yes
Resistance server impersonating attack	No	No	Yes	Yes	Yes
Resistance of replay attack	Yes	Yes	Yes	Yes	Yes
Provision user anonymity	No	No	No	No	Yes
Achieving mutual authentication	Yes	Yes	Yes	Yes	Yes
Provision of perfect forward secrecy	Yes	Yes	No	Yes	Yes
Session key establishment	Yes	Yes	Yes	Yes	Yes

Table 3: Performance comparisons

	Type of operation	[15]	[22]	[10]	[20]	ours
Computation cost	T_H	9	14	6	12	7
	T_{asy}	4	5	6	4	7

card, he/she also cannot obtain X_i without knowing ID_i and PW_i. Hence, the adversary cannot impersonate as a legal user to login S by launching the forgery attack.

3.2.5 Server Impersonating Attack

Suppose that the adversary intercepts the login request of U_i, and responds a forged reply message to impersonate servers. However, he/she cannot obtain the key information $ID_i \cdot P$ and $a \cdot b \cdot P$ without knowing the master secret key x which kept only by S, even if the attacker has extracted the data from U_i's smart card. U_i can verify V_s to authenticate S, which is contributed with theses two key information. Hence, the attacker cannot masquerade as S to fool U_i by launching the server impersonating attack.

3.2.6 Known Key Attack

Due to the session key SK is computed by two random numbers which employs the principle of Diffi-Hellman key exchange protocol. Thus neither the value of session key SK is the same with any other authentication message, nor SK as a part of any other authentication message. The leakage of SK does not affect other unexposed sessions. Thus, the known key attack is resisted effectively.

3.2.7 Reply Attack

The replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. If the adversary can eavesdrop any request from U_i to S, and replays the intercepted one to S. Firstly, S can reject mostly session by verifying the freshness of timestamp. Secondly, even if the login request passes the verification of timestamp, S will return R_s, V_s, T_s with contribution of random number b. the attacker also cannot obtain a which is implied in the login request to compute the session key SK. Hence, our scheme can effectively resist replay attack.

4 Performance Analysis

In this section, we compare the security features and efficiency with some existing state-of-the-art such as: Lee *et al.*'s [15], Xue *et al.*'s [22], Jiang *et al.*'s and Wei *et al.*'s schemes to evaluating our scheme. And the comparative summary of the proposed scheme is presented in the Table 2. It is clear that only our proposed scheme can achieve all requirements list in Table2. Specifically, the innovative feature is that we prove the completeness of our scheme by employing BAN-logic.

Table 3 summarizes the comparison among our scheme and the aforementioned related works in terms of efficiency. We define the T_H indicates the time complexity of hash function and T_{asy} is the time complexity of the asymmetric encryption. The computations cost of Lee *et al.*'s [15], Xue *et al.*'s [22], Jiang *et al.*'s [10] and Wei *et al.*'s [20] schemes are $9T_H + 4T_{asy}$, $14T_H + 4T_{asy}$, $6T_H + 6T_{asy}$, $12T_H + 4T_{asy}$. Our schemes need $7T_H + 7T_{asy}$ which slightly needs more computational cost than other schemes in asymmetric encryption operation. However, our scheme can achieve formal proof security and more admired security properties.

5 Conclusions

In this paper, we propose a secure authentication scheme using elliptic curves cryptosystem which could conquer a range of potential network attacks. Furthermore, the formal proof based on BAN-logic shows that the validity of the proposed scheme. The performance analysis indicates that our proposed scheme is relatively more robust than the related schemes, while increasing little computational efficiency.

Acknowledgments

This study was supported by the Nanyang Normal University Project under grant QN2015008. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] D. S. AbdElminaam, H. M. Abdul Kader, M. M. Hadhoud, S. M. El-Sayed, "Increase the Performance of Mobile Smartphones using Partition and Migration of Mobile Applications to Cloud Computing," *International Journal of Electronics and Information Engineering*, vol. 1, no. 1, pp. 34-44, 2014.

[2] A. K Awasthi, "Comment on a dynamic ID-based remote user authentication scheme," *Transactions on Cryptology*, vol. 1, no. 2, pp. 15-16, 2004.

[3] M. Burrows, M. Abadi, R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18-36, 1990.

[4] C. C. Chang, C. Y. Lee, "A smart card-based authentication scheme using user identify cryptography," *International Journal of Network Security*, vol. 15, no. 2, pp. 139-147, 2013.

[5] M. L. Das, A. Saxena, V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no.2, pp. 629-631, 2004.

[6] D. L. Guo, F. T. Wen, "A more robust authentication scheme for roaming service in global mobility networks using ECC," *International Journal of Network Security*, vol. 18, no. 2, pp. 217-233, 2016.

[7] D. Hankerson, A. Menezes, S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer Science & Business Media, 2006.

[8] M. S. Hwang, C. C. Lee, S. K. Chong, J. W. Lo, "A Key Management for Wireless Communications," *International Journal of Innovative Computing, Information and Control*, Vol. 4, No. 8, pp. 2045-2056, 2008.

[9] D. B. He, J. H. Chen, R. Zhang, "Weaknesses of a dynamic ID-based remote user authentication scheme," *International Journal of Electronic Security and Digital Forensics*, vol. 3, no. 4, pp. 355-362, 2010.

[10] Q. Jiang, J. Ma, G. Li, X. Li, "Improvement of robust smart-card-based password authentication scheme," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 383-393, 2015.

[11] M. K. Khan, S. K. Kim, K. Alghathbar, "Cryptanalysis and security enhancement of a 'more efficient & secure dynamic ID-based remote user authentication scheme'," *Computer Communications*, vol. 34, no. 3, pp. 305-309, 2011.

[12] P. Kocher, J. Jaffe, B. Jun, "Differential power analysis," *Proceedings of Advances in Cryptology*, Santa Barbara, CA, U.S.A., pp. 388-397, 1999.

[13] N. Koblitz, "Elliptic curve cryptosystem," *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, 1987.

[14] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770-772, 1981.

[15] C. Lee, C. Chen, C. Wu, S. Huang, "An extended chaotic maps-based key agreement protocol with user anonymity," *Nonlinear Dynamics*, vol. 69, no. 1-2, pp. 79-87, 2012.

[16] X. Li, J. W. Niu, M. K. Khan, J. G. Liao, "An enhanced smart card based remote user password authentication scheme," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1365-1371, 2013.

[17] Y. Lin, "Chaotic map based mobile dynamic ID authenticaed key agreement scheme," *Wireless Personal Communications*. vol. 78, no. 2, pp. 1487-1494, 2014.

[18] T. S. Messerges, E. A. Dabbish, R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computer*, vol. 5, no. 51, pp. 541-552, 2002.

[19] Y. Y. Wang, J. Y. Liu, F. X. Xiao, J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 4, no. 32, pp. 583-585, 2009.

[20] J. H. Wei, W. F. Liu, X. X. Hu, "Secure and efficient smart card based remote user password authentication scheme," *International Journal of Network Security*, vol. 18, no. 4, pp. 782-792, 2016.

[21] D. Xiao, X. Liao, S. Deng, "A novel key agreement protocol based on chaotic maps," *Information Sciences*, vol. 177, no. 4, pp. 1136-1142, 2007.

[22] K. Xue, P. Hong, "Security improvement on an anonymious key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 7, pp. 2969-2977, 2012.

Xueqin Zhang received the B.S. and M.S. degrees in Computer Science and Technology from Henan Polytechnic University, Jiaozuo, Henan, China in 2005 and 2008, respectively. She is a lecturer in Nanyang Normal University, Nanyang, Henan, China. Her research interests include data mining and information security.

Baoping Wang received B.S. degree in Computer Application Technology from Henan University, Kaifeng, China in 1997 and M.S. in Computer Application Technology from Guizhou University, Guiyang, China in 2006. He is an associate professor in Nanyang Normal University, Henan, China. His research interests include computer network technology and information security.

Wei Wang received the B.S. and M.S. degrees in Computer Science and Technology from Northwest agriculture and forestry university of science and technology, Yangling, Shanxi, China in 2003 and 2006, respectively. He is a lecturer in Nanyang Normal University, Nanyang, Henan, China. Her research interests include data mining and information security.

A Secure ECC-based Mobile RFID Mutual Authentication Protocol and Its Application

Shin-Yan Chiou, Wen-Tsai Ko, and Erl-Huei Lu
(Corresponding author: Shin-Yan Chiou)

Department of Electrical Engineering, Chang Gung University
259, Wen-Hwa 1st Road, Kwei-Shan, Tao-Yuan, Taiwan
(Email: corresponding_ansel@mail.cgu.edu.tw)

(Received Feb. 15, 2017; revised and accepted June 25, 2017)

Abstract

Mobile RFID applications combine RFID technologies and mobile device to create a new convenient application area. However, most of the applications suffer from the security issues due to insecure communication channels among tags, readers and servers. In 2012, Zhou et al. proposed an ECC-based mutual authentication protocol to promote mobile RFID applications security. However, we found their protocol faces to OTRUTS (one time reading and unlimited times service) problem, which means once a reader read the data of a certain tags from a server successfully, the reader can read it unlimited times without reading those tags again. Therefore, their protocol cannot securely support some mobile RFID applications such as the security patrolling application. In this paper, we propose a new secure ECC-based mobile RFID mutual authentication protocol for the safety of mobile RFID applications such as security patrolling.

Keywords: ECC; Mutual Authentication; RFID

1 Introduction

Recently, many researches have concentrated on mobile RFID-based applications [1, 2, 4, 12, 19, 23, 24, 26, 27, 29, 30] as it is believed that this type of applications have advantages of both RFID technology and mobile smart device. In most of the traditional RFID applications, it is assumed that the communication between reader and back end server is wired and secure, while it between tag and reader is wireless and insecure. This is because readers are usually installed at a fixed location but the tags are mobile.

However, in mobile RFID applications, both tag-reader and reader-backend server communication channels are in wireless transmission mode and therefore considered to be insecure. In the mobile RFID based telecommunication service, the tags (different from those of traditional RFID applications) are designed to be stationary and the readers (installed in a mobile device such as a cell phone) are movable. The mobile RFID telecommunication services provide tag information which is stored and maintained in backend database over a reader embedded mobile network to support many applications such as mobile payment [5, 11, 18, 21], emergency response [6, 15, 20, 23], marketing [2], advertisements promotion [14], security patrolling, position reporting, etc. Therefore, the mobile device, with an embedded reader, could be used by a potential customer, a consumer, a security patrolman, etc. That means the holder of the mobile device could also be an adversary to the mobile RFID system.

ECC is proved to be suitable for RFID applications [3, 7, 8, 10, 13, 17, 22, 25]. In 2012, Zhou *et al.* [28] proposed a mutual authentication protocol based on public-key cryptography using ECC for mobile RFID application.

However, their protocol has OTRUTS (one time reading and unlimited times service) problem, which means once a reader read the data of a certain tags from a server successfully, the reader can read it unlimited times without reading those tags again. Therefore, their protocol cannot securely support some mobile RFID applications such as the security patrolling application. In this paper, we propose a new secure ECC-based mobile RFID mutual authentication protocol for the safety of mobile RFID applications such as security patrolling.

The rest of this paper is organized as follows. Second section provides a brief background of ECC. In the third and forth section, we review and analyze Zhon *et al.*'s protocol. The proposed scheme is demonstrated in fifth section. Sixth section provides security analysis. Finally, we draw conclusions in seventh section.

2 Preliminaries

For ECC application, a non-singular elliptic curve should be chosen. All points in a non-singular elliptic curve ($y^2 = x^3 + ax + b(\text{mod } p)$) have tangent lines except one point at infinity, where $p > 3$ and $4a^3 + 27b^2 \neq 0$. The security of ECC is based on the intractability of the following problems.

2.1 Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given an elliptic curve E defined over a finite field F_q, denoted by $E(F_q)$. There is a point $P \in E(F_q)$ with order m and a point $Q \in< P >$. Then the problem of finding the integer $k \in [0, m-1]$ from given P and Q such that $Q = kP$ is defined as ECDLP, where k is the discrete logarithm of Q to the base P, denoted $k = \log_P Q$ [9].

2.2 Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP)

From three given points P, xP and yP over $E(F_q)$, it is hard to compute xyP over $E(F_q)$.

2.3 Elliptic Curve Factorization Problem(ECFP)

From two given points P and Q over $E(F_q)$, where $Q = xP + yP$, it is hard to find two points xP and yP over $E(F_q)$ [16].

3 Zhou *et al.*'s Protocol

In this section, we provide a brief introduction to the notations and Zhou *et al.*'s protocol. Table 1 shows the notations used in our and Zhou *et al.*'s scheme. In table 1, $h()$ is an one-way hash function, where $h : \{0, 1\}^* \to \{0, 1\}^{2m}$, m is the bit length of the coordinate x or y of a point over the elliptic curve $E(F_q)$.

Table 1: Notations

Notation	Meaning
P	a base point of a subgroup on elliptic curve $E(F_q)$
ID_i	the identity of Tag_i
$\bar{x}(T)$	the x-coordinate of a point $T = (x_t, y_t)$
$\bar{y}(T)$	the y-coordinate of a point $T = (x_t, y_t)$
t_R	time
k_1, k_2	secrets of *Server*
k_3	private key of *Server*
K	public key of *Server*
T_{P_i}	pseudo-id of Tag_i
R_P	public key of *Reader*
r	private key of *Reader*
$h(\,)$	one-way hash function
$(S)_L$	the left half bits of a binary sequence S
$(S)_R$	the right half bits of a binary sequence S

Zhou *et al.*'s protocol (Figure 1) has four phases: (1) *Initialization Phase*, (2) *Mobile Reader Authentication Phase*, (3) *Tag Authentication Phase* and (4) *Tag Information Sending Phase*. Those phases are described as follows.

3.1 Initialization Phase

In this phase, *Server* chooses an elliptic curve $E(F_q)$ and a base point P over $E(F_q)$ with order n, where n is a large prime number. *Server* randomly chooses his secrets k_1 and $k_2 \in_R Z_q^*$ and his private key $k_3 \in_R Z_q^*$, and computes his public key $K = k_3P$. Next, *Server* computes pseudo-ids $T_{P_i} = k_1^{-1}ID_i + k_2P$ for each tag and writes T_{P_i} into Tagi's memory. On the other hand, Reader randomly chooses his private key $r \in_R Z_q^*$ and computes his public key $RP = rP$.

3.2 Mobile Reader Authentication Phase

In this phase, *Reader* randomly chooses $s \in_R Z_q^*$, computes $Q = sP$, and sends a request Q to Tag_i. After Tag_i receives Q, it chooses a random number $t \in_R Z_q^*$ and sends t to *Reader*. Next, *Reader* computes $v = rt - s$ and sends v to Tag_i. Tag_i checks whether $vP + Q$ is equal to tR_P. If it does, Tag_i authenticates the *Reader* successfully. Otherwise, Tag_i aborts this communication.

3.3 Tag Authentication Phase

In this phase, the Tagi first chooses a random number $c \in_R Z_q^*$, computes $T_1 = cP$, $T_2 = cQ$, $T_3 = cK$, $T_4 = T_{P_i} + T_3$ and $u = h(\bar{x}(T_2), \bar{y}(T_4))$, and sends T_1, T_4 and u to *Reader*. After *Reader* receives them, it computes $R_1 = sT_1$ and $w = h(\bar{x}(R_2), \bar{y}(T_4))$, and checks whether w is equal to u. If it does not, *Reader* aborts this session. Otherwise, *Reader* considers T_1, T_4 and u as valid parameters. Next, *Reader* chooses a random number g, extracts time t_R, computes $R_2 = gP$, $R_3 = (r + g)K$ and $d_R = h(\bar{y}(R_3), \bar{x}(T_1), \bar{x}(T_4), t_R)$, and sends T_1, T_4, R_2, t_R and d_R to *Server*. After *Server* receives those messages, it checks whether t_R is valid. If it does, *Server* computes $B_1 = k_3(R_P + R_2)$ and $d_B = h(\bar{y}(B_1), \bar{x}(T_1), \bar{x}(T_4), t_R)$. Otherwise, *Server* aborts this session. *Server* checks whether $d_B = d_R$ holds, and considers T_1, T_4, R_2, t_R and d_R as valid parameters and authenticates *Reader* successfully. Next, *Server* computes $ID_i = k_1(B_2 - k_2P)$ and checks whether ID_i exists in the database. If it does, *Server* authenticates Tag_i successfully. Otherwise, *Server* aborts this session.

3.4 Tag Information Sending Phase

In this phase, *Server* fetches the related $DATA_i$ of ID_i from the database, encrypts it, and sends the encrypted data to *Reader*. *Server* first chooses a random number $l \in_R Z_q^*$, computes $B_3 = lP$, $B_4 = lR_P$, $B_5 = k_3R_P$, $d_1 = \bar{y}(B_4) \oplus (DATA_i)_L || \bar{x}(B_5) \oplus (DATA_i)_R$ and $d_2 =$

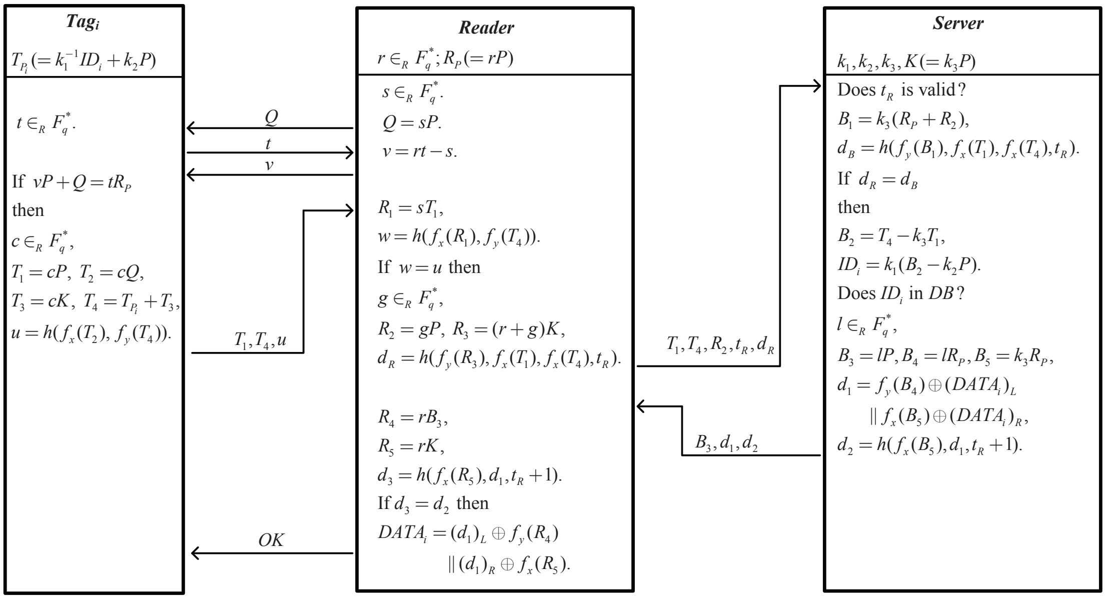


Figure 1: Zhou *et al.*'s protocol

$h(\bar{x}(B_5), d_1, t_R + 1)$, and sends B_3, d_1 and d_2 to $Reader$. When $Reader$ receives those messages, it computes $R_4 = rB_3$, $R_5 = rK$ and $d_3 = h(\bar{x}(R_5), d_1, t_R + 1)$, and checks whether $d_3 = d_2$ holds. If it does, $Reader$ believe that the parameters B_3, d_1 and d_2 are sent from $Server$, and recovers $DATA_i = (d_1)_L \oplus \bar{y}(R_4)||(d_1)_R \oplus \bar{x}(R_5)$. Otherwise, $Reader$ aborts this session.

4 Analysis on Zhou *et al.*'s Protocol

In Zhou *et al.*'s protocol, we find that once $Reader$ has read Tag_i's data, then the $Reader$ can get Tag_i's data from $Server$ without reading Tag_i again. We name this problem as "One Time Reading, Unlimited Times Service (OTRUTS)." The detail of this problem is described as follows and shown in Figure 2.

Assume $Reader$ read Tag_i once get the data of Tag_i from $Server$ successfully. $Reader$ have valid T_1 and T_4. As $Reader$ wants Tag_i's new $DATA_i$ from $Server$, he can assign $T'_1 = T_1$, $T'_2 = T_2$, extract a new time t'_R, generate a random number$g' \in_R Z_q^*$, compute $R'_2 = g'P$,$R'_3 = (r + g')K$ and $d'_R = h(\bar{y}(R'_3), \bar{x}(T'_1), \bar{x}(T'_4), t'_R)$, and send $T'_1, T'_4, R'_2, t'_R, d'_R$to $Server$ to request the $DATA_i$ of Tag_i. As $Server$ received those messages from $Reader$, $Server$ authenticates t'_R successfully (with no doubt), computes $B'_1 = k_3(R_P + R'_2)$ and $d'_B = h(\bar{y}(B'_1), \bar{x}(T'_1), \bar{x}(T'_4), t'_R)$, finds out $d'_B = d'_R$ holds, and computes $ID_i = k_1(B'_2 - k_2P)$. Thus $Server$ can successfully find ID_i in database because B'_2 has the pseudo-id information T_{P_i}. Therefore, $Server$ can fetch the $DATA_i$ and process the "$TagInformationSendingPhase$" to encrypt $DATA_i$ for $Reader$'s request. $Server$ then generates a random number $l' \in_R Z_q^*$, computes $B'_3 = l'P$, $B'_4 = l'R_P$, $B'_5 = k_3R_P$, $d'_1 = \bar{y}(B'_4) \oplus (DATA_i)_L||\bar{x}(B'_5) \oplus (DATA_i)_R$ and $d'_2 = h(\bar{x}(B'_5), d'_1, t'_R + 1)$, and sends $B'_3, d'_1$and d'_2 to $Reader$. After $Reader$ receives those message, it computes $R'_4 = rB'_3$ and$R'_5 = rK$.Thus, $Reader$ can recover $DATA_i = (d'_1)_L \oplus \bar{y}(R'_4)||(d'_1)_R \oplus \bar{x}(R'_5)$. Therefore, in Zhou *et al.*'s protocol, $Reader$ just needs to read Tag_i's $DATA_i$ one time, then he can read Tag_i's $DATA_i$ from $Server$ with unlimited times without reading Tag_i again.

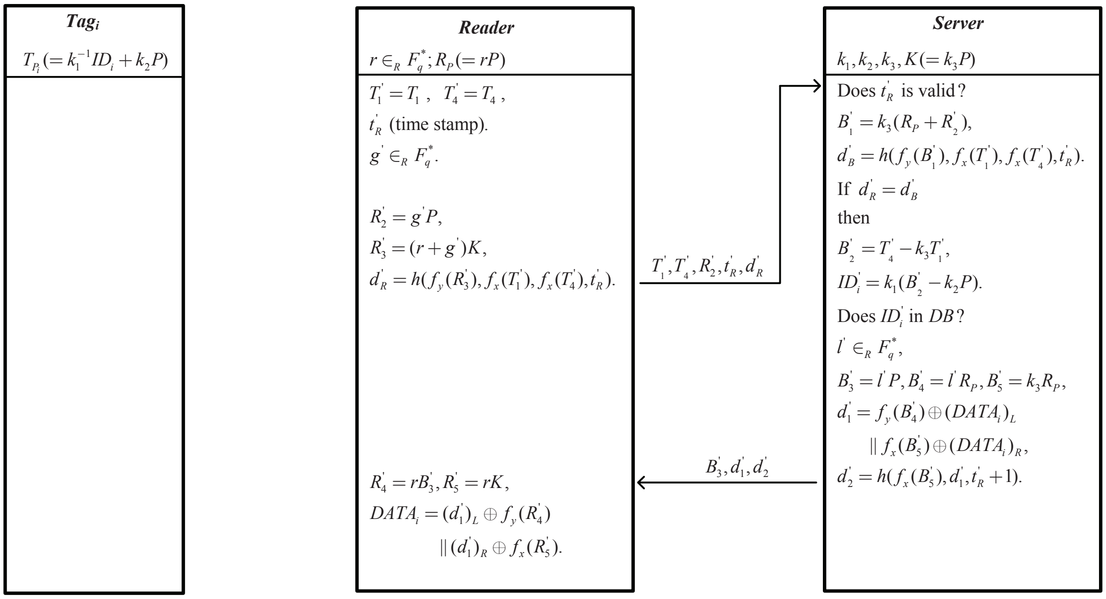
5 Proposed Protocol

In this section, we propose a mobile RFID-based mutual authentication protocol using elliptic curve cryptography for security patrolling application. In our protocol, we fix the OTRUTS problem of Zhou *et al.*'s protocol and make our protocol suitable to secure applications such as security patrolling.

We take a security patrolling scenario as an instance. In the security patrolling scenario, there are three roles: (1) $Server$ as the Security Management Center (SMC), (2) $Reader$ as the patrolman's $Reader$ (PMR), and (3)Tag_i as the sentry post's Tag_i (SPT_i). Our protocol has four phases: (1) *Initialization Phase*, (2) *SPT_i to PMR Authentication Phase*, (3) *SMC to PMR and SPT_i Authentication Phase* and (4) *$DATA_i$ Sending Phase*. These phases are described as follows and shown in Figure 3.

5.1 Initialization Phase

The initialization phase is same as Zhou *et al.*'s protocol. SMC chooses an elliptic curve $E(F_q)$ and a base point P over $E(F_q)$ with order n, where n is a large prime number. SMC chooses two secrets k_1, $k_2 \in_R Z_q^*$ and one private



Tag$_i$
$T_{P_i}(=k_1^{-1}ID_i+k_2P)$
Reader
$r\in_R F_q^*;R_P(=rP)$
$T_1'=T_1$, $T_4'=T_4$,
t_R' (time stamp).
$g'\in_R F_q^*$.
$R_2'=g'P$,
$R_3'=(r+g')K$,
$d_R'=h(f_y(R_3'),f_x(T_1'),f_x(T_4'),t_R')$.
T_1',T_4',R_2',t_R',d_R'
Server
$k_1,k_2,k_3,K(=k_3P)$
Does t_R' is valid?
$B_1'=k_3(R_P+R_2')$,
$d_B'=h(f_y(B_1'),f_x(T_1'),f_x(T_4'),t_R')$.
If $d_R'=d_B'$
then
$B_2'=T_4'-k_3T_1'$,
$ID_i'=k_1(B_2'-k_2P)$.
Does ID_i' in DB?
$l'\in_R F_q^*$,
$B_3'=l'P,B_4'=l'R_P,B_5'=k_3R_P$,
$d_1'=f_y(B_4')\oplus(DATA_i)_L$
$\| f_x(B_5')\oplus(DATA_i)_R$,
$d_2'=h(f_x(B_5'),d_1',t_R'+1)$.
B_3',d_1',d_2'
$R_4'=rB_3',R_5'=rK$,
$DATA_i=(d_1')_L\oplus f_y(R_4')$
$\| (d_1')_R\oplus f_x(R_5')$.

Figure 2: OTRUTS problem

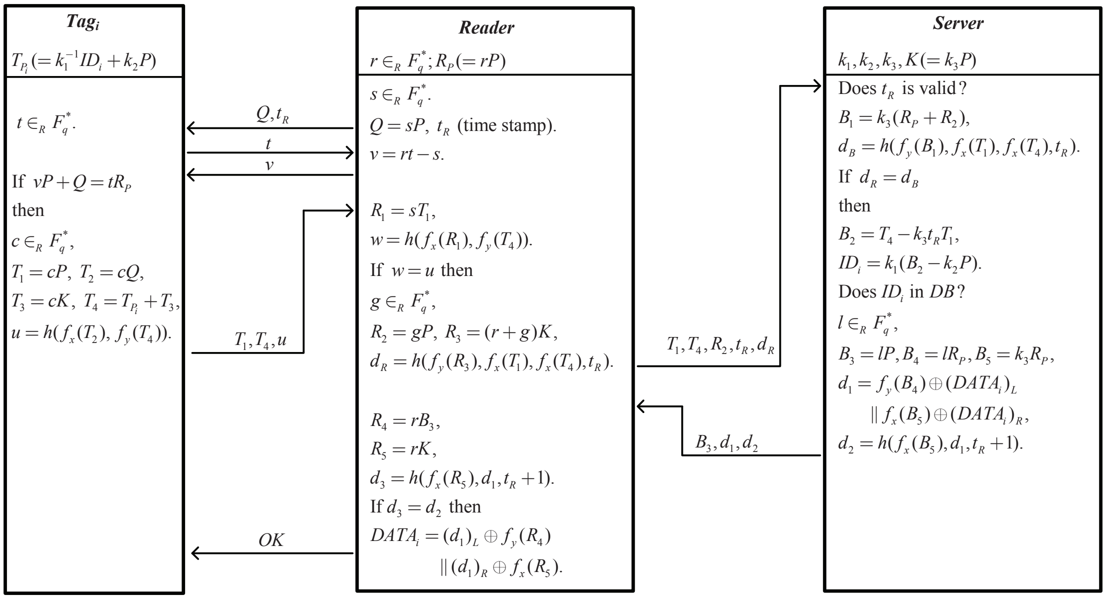


Figure 3: Proposed protocol

key $k_3 \in_R Z_q^*$, and computes his public key $K = k_3P$. Each tag has pseudo-id $T_{P_i} = k_1^{-1}ID_i + k_2P$. On the other hand, PMR chooses his private key $r \in_R Z_q^*$ computes his public key $R_P = rP$.

5.2 SPT_i to PMR Authentication Phase

In this phase, PMR randomly chooses $s \in_R Z_q^*$, extracts times t_R and computes $Q = sP$. Then PMR sends a request, Q and t_R, to SPT_i. After SPT_i receives Q and t_R, it randomly chooses $t \in_R Z_q^*$ and replies t to PMR. After PMR receives t, it computes $v = rt - s$ and sends v to SPT_i. SPT_i checks whether $vP + Q = tR_P$ holds. If it does, SPT_i authenticates the PMR successfully. Otherwise, it aborts the communication.

5.3 SMC to PMR and SPT_i Authentication Phase

SPT_i randomly chooses $c \in_R Z_q^*$, computes $T_1 = cP$, $T_2 = cQ$, $T_3 = ct_RK$, $T_4 = T_{P_i} + T_3$ and $u = h(\bar{x}(T_2), \bar{y}(T_4))$, and sends T_1, T_4 and u to PMR. After PMR receives T_1, T_4 and u, it computes $R_1 = sT_1$ and $w = h(\bar{x}(R_2), \bar{y}(T_4))$, and checks whether $w = u$. If it does, PMR authenticates the messages T_1, T_4 and u successfully. Otherwise, it aborts this session. Then PMR chooses a random number g$\in_R Z_n^*$, computes $R_2 = gP$, $R_3 = (r + g)K$ and $d_R = h(\bar{y}(R_3), \bar{x}(T_1), \bar{x}(T_4), t_R)$, and sends T_1, T_4, R_2, t_R and d_R to $Server$. Then $Server$ checks whether t_R is valid. If it does not, $Server$ aborts this session. Otherwise, $Server$ computes $B_1 = k_3(R_P + R_2)$ and $d_B = h(\bar{y}(B_1), \bar{x}(T_1), \bar{x}(T_4), t_R)$, and checks whether $d_B = d_R$ holds. If it does, $Server$ considers T_1, T_4, R_2, t_R and d_R as valid parameters and authenticate $Reader$ successfully. Next, $Server$ computes $B_2 = T_4 - k_3t_RT_1$and $ID_i = k_1(B_2 - k_2P)$, and checks whether ID_i exists in the database. If it does, $Server$ authenticate Tag_i successfully. Otherwise, $Server$ aborts this session.

5.4 $DATA_i$ Sending Phase

In this phase, $Server$ fetches the related $DATA_i$ of ID_i from the database, encrypts it, and sends the encrypted data to PMR. First, SMC randomly chooses $l \in_R Z_q^*$, computes $B_3 = lP$, $B_4 = lR_P$, $B_5 = k_3R_P$, $d_1 = \bar{y}(B_4) \oplus (DATA_i)_L || \bar{x}(B_5) \oplus (DATA_i)_R$ and $d_2 = h(\bar{x}(B_5), d_1, t_R{+}1)$, and sends B_3, d_1 and d_2 to PMR. After PMR receives those messages, it computes $R_4 = rB_3$, $R_5 = rK$, $d_3 = h(\bar{x}(R_5), d_1, t_R + 1)$, and checks whether $d_3 = d_2$ holds. If it does, PMR believes the parameters B_3, d_1 and d_2 comes from a valid SMC, and recovers $DATA_i = (d_1)_L \oplus \bar{y}(R_4) || (d_1)_R \oplus \bar{x}(R_5)$. Otherwise, it aborts this session.

6 Security Analysis

In the security patrolling scenario, PMR is supposed to visit the assigned SPT in person, read the SPT and send proof back to the SMC for verification in a valid time interval. If a protocol has the OTRUTS problem (described in section 3), PMR just needs to visit SPT_i only one time then he can sit on the chair in the security office and complete the patrolling report without visiting the same SPT again. Therefore, a security patrolling application should avoid the OTRUTS problem in the RFID mobile mutual authentication protocol.

In our protocol, we rearranged $T_3 = ct_RK$ to solve this OTRUTS problem. Thus, we have $T_4 = T_{P_i} + ct_RK$. If PMR tries to read SPT_i's data from SMC without reading SPT_i again, shown as Figure 4, he assigns $T_1' = T_1$ and $T_2' = T_2$, extracts a new time t_R', generates a random number $g' \in_R Z_n^*$, computes $R_2' = g'P$, $R_3' = (r + g')K$ and $d_R' = h(\bar{y}(R_3'), \bar{x}(T_1'), \bar{x}(T_4'), t_R')$, and sends $T_1', T_4', R_2', t_R', d_R'$ to SMC. After SMC receives these messages, it authenticates t_R' successfully (with no doubt), computes $B_1' = k_3(R_P + R_2')$ and $d_B' = h(\bar{y}(B_1'), \bar{x}(T_1'), \bar{x}(T_4'), t_R')$, finds $d_B' = d_R'$ holds, and compute $B_2' = T_4' - k_3t_R'T_1' = T_{P_i} + (t_R - t_R')ck_3P$. Now SMC tries to recover ID_i by computing $ID_i' = k_1(B_2' - k_2P)$ $= k_1(T_{P_i} + (t_R - t_R')ck_3P - k_2P) = ID_i + (t_R - t_R')ck_3P$ $\neq ID_i$. However, SMC finds out ID_i' is not in the database and aborts the session. Therefore, our protocol not only provides the security properties of Zhou *et al.*'s protocol, but also resistants to OTRUTS problem which make our protocol more suitable for security patrolling application.

7 Conclusions

This paper discusses the Zhou *et al.*'s mutual authentication protocol and points out their protocol is faces OTRUTS problem and therefore cannot securely support some mobile RFID applications such as the security patrolling application. This paper proposes a new mutual authentication using ECC and proved the proposed protocol is resistant to OTRUTS problem.

Acknowledgments

This work is partially supported by the Ministry of Science and Technology under Grant MOST 104-2221-E-182-012 and by the CGMH project under Grant BMRPB46. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] D. Benedetti, G. Maselli, and C. Petrioli, "Prime: Priority-based tag identification in mobile RFID sys-

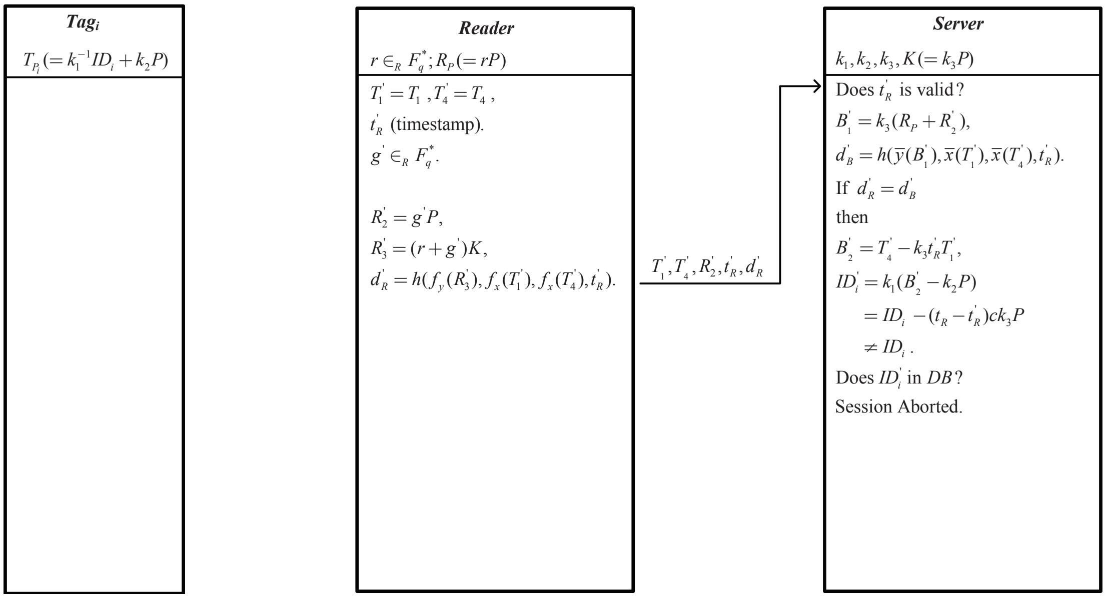

Tag$_i$
$T_{P_i}(=k_1^{-1}ID_i+k_2P)$
Reader
$r\in_R F_q^*; R_P(=rP)$
$T_1^{'}=T_1, T_4^{'}=T_4,$
$t_R^{'}$ (timestamp).
$g^{'}\in_R F_q^*.$
$R_2^{'}=g^{'}P,$
$R_3^{'}=(r+g^{'})K,$
$d_R^{'}=h(f_y(R_3^{'}),f_x(T_1^{'}),f_x(T_4^{'}),t_R^{'}).$
$T_1^{'},T_4^{'},R_2^{'},t_R^{'},d_R^{'}$
Server
$k_1,k_2,k_3,K(=k_3P)$
Does $t_R^{'}$ is valid?
$B_1^{'}=k_3(R_P+R_2^{'}),$
$d_B^{'}=h(\overline{y}(B_1^{'}),\overline{x}(T_1^{'}),\overline{x}(T_4^{'}),t_R^{'}).$
If $d_R^{'}=d_B^{'}$
then
$B_2^{'}=T_4^{'}-k_3t_R^{'}T_1^{'},$
$ID_i^{'}=k_1(B_2^{'}-k_2P)$
$=ID_i-(t_R-t_R^{'})ck_3P$
$\neq ID_i.$
Does $ID_i^{'}$ in DB?
Session Aborted.

Figure 4: The resistance of OTRUTS problem in our protocol

tems," *Computer Communications*, vol. 108, pp. 64-77, 2017.

[2] C. L. Chen, J. K. Jan, and C. F. Chien, "Based on mobile RFID device to design a secure mutual authentication scheme for market application," in *International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA'10)*, pp. 423–428, 2010.

[3] Y. Chen and J. S. Chou, "Ecc-based untraceable authentication for large-scale active-tag RFID systems," *Electronic Commerce Research*, vol. 15, no. 1, pp. 97, 2015.

[4] P. Y. Cui, "An improved ownership transfer and mutual authentication for lightweight RFID protocols," *International Journal of Network Security*, vol. 18, no. 6, pp. 1173–1179, 2016.

[5] T. Falk, W. H. Kunz, J. J. Schepers, and A. J. Mrozek, "How mobile payment influences the overall store price image," *Journal of Business Research*, vol. 69, no. 7, pp. 2417–2423, 2016.

[6] G. L. Foresti, M. Farinosi, and M. Vernier, "Situational awareness in smart environments: socio-mobile and sensor data fusion for emergency response to disasters," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 6, pp. 239–257, 2015.

[7] D. Guo and F. Wen, "A more robust authentication scheme for roaming service in global mobility networks using ECC," *International Journal of Network Security*, vol. 18, no. 2, pp. 217-223, 2016.

[8] L. Han, Q. Xie, and W. Liu, "An improved biometric based authentication scheme with user anonymity using elliptic curve cryptosystem," *International Journal of Network Security*, vol. 19, no. 3, pp. 469-478, 2017.

[9] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer Science & Business Media, 2006.

[10] G. Hou, Z. Wang, "A robust and efficient remote authentication scheme from elliptic curve cryptosystem," *International Journal of Network Security*, vol. 19, no. 6, pp. 904-911, 2017.

[11] G. De Kerviler, N. T. Demoulin, and P. Zidda, "Adoption of in-store mobile payment: Are perceived risk and convenience the only drivers?" *Journal of Retailing and Consumer Services*, vol. 31, pp. 334–344, 2016.

[12] N. Kumar, R. Iqbal, S. Misra, J. J. Rodrigues, and M. S. Obaidat, "Bayesian cooperative coalition game as-a-service for RFID-based secure QOS management in mobile cloud," *IEEE Transactions on Emerging Topics in Computing*, 2016.

[13] N. Kumar, K. Kaur, S. C. Misra, and R. Iqbal, "An intelligent RFID-enabled authentication scheme for healthcare applications in vehicular mobile cloud," *Peer-to-Peer Networking and Applications*, vol. 5, no. 9, pp. 824–840, 2016.

[14] K. Y. Lam, J. K. Y. Ng, J. Wang, et al., "A pervasive promotion model for personalized promotion systems on using wlan localization and nfc techniques," *Mobile Information Systems*, vol. 2015, 2015.

[15] K. M. Lee, M. Runyon, T. J. Herrman, R. Phillips, and J. Hsieh, "Review of salmonella detection and identification methods: Aspects of rapid emergency response and food safety," *Food Control*, vol. 47, pp. 264–276, 2015.

[16] F. Li, X. Xin, and Y. Hu, "Indentity-based broadcast signcryption," *Computer Standards & Interfaces*, vol. 30, no. 1, pp. 89–94, 2008.

[17] D. Liu, Z. Liu, Z. Yong, X. Zou, and J. Cheng, "Design and implementation of an ecc-based digital baseband controller for RFID tag chip," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 7, pp. 4365–4373, 2015.

[18] T. Oliveira, M. Thomas, G. Baptista, and F. Campos, "Mobile payment: Understanding the determinants of customer adoption and intention to recommend the technology," *Computers in Human Behavior*, vol. 61, pp. 404–414, 2016.

[19] Q. Qian, Y. L. Jia, R. Zhang, "A lightweight RFID security protocol based on elliptic curve Cryptography," *International Journal of Network Security*, vol. 18, no. 2, pp. 354–361, 2016.

[20] J. Salerno, W. M. Hlaing, T. Weiser, C. Striley, L. Schwartz, F. J. Angulo, and V. S. Neslund, "Emergency response in a global health crisis: epidemiology, ethics, and ebola application," *Annals of Epidemiology*, vol. 26, no. 4, pp. 234–237, 2016.

[21] A. A. Shaikh, P. Hanafizadeh, and H. Karjaluoto, "Mobile banking and payment system: A conceptual standpoint," *International Journal of E-Business Research*, vol. 13, no. 2, pp. 14–27, 2017.

[22] X. Tan, M. Dong, C. Wu, K. Ota, J. Wang, and D. W. Engels, "An energy-efficient ecc processor of uhf RFID tag for banknote anti-counterfeiting," *IEEE Access*, vol. 5, pp. 3044–3054, 2017.

[23] T. Tran, F. Z. Yousaf, and C. Wietfeld, "Rfid based secure mobile communication framework for emergency response management," in *IEEE Wireless Communications and Networking Conference*, pp. 1–6, 2010.

[24] B. Vaidya, D. Makrakis, and H. T. Mouftah, "Authentication mechanism for mobile RFID based smart grid network," in *IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE'14)*, pp. 1–6, 2014.

[25] J. Wolkerstorfer, "Is elliptic-curve cryptography suitable to secure RFID tags," in *Handout of the Ecrypt Workshop on RFID and Lightweight Crypto*, 2005.

[26] R. Xie, B. Y. Jian, and D. W. Liu, "An improved ownership transfer for RFID protocol," *International Journal of Network Security*, vol. 20, no. 1, pp. 149-156, 2018.

[27] L. Yang, Y. Chen, X. Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile RFID tags to high precision using cots devices," in *Proceedings of ACM 20th Annual International Conference on Mobile Computing and Networking*, pp. 237–248, 2014.

[28] J. Zhou, Y. Zhou, F. Xiao, and X. Niu, "Mutual authentication protocol for mobile RFID systems," *Journal of Computational Information Systems*, vol. 8, no. 8, pp. 3261–3268, 2012.

[29] Y. Zou, J. Xiao, J. Han, K. Wu, Y. Li, and L. M. Ni, "GRFID: A device-free RFID-based gesture recognition system," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 381–393, 2017.

[30] J. M. Zydney and Z. Warner, "Mobile apps for science learning: Review of research," *Computers & Education*, vol. 94, pp. 1–17, 2016.

Biography

Shin-Yan Chiou received the PhD degree in Electrical Engineering from National Cheng Kung University, Taiwan, in 2004. From 2004 to 2009, he worked at Industrial Technology Research Institute as a RD Engineer. Since 2009, he joined the faculty of the Department of Electrical Engineering, Chang Gung University, Taoyuan, Taiwan, where he is currently an Associate Professor. His research interests include information security, cryptography, social network security, and secure applications between mobile devices.

Wen-Tsai Ko received the B.S. degree in Applied Mathematics from Chung Cheng Institute of Technology in 1986, the M.B.A. degree in Defense Information from National Defense Management College in 1998, and the PhD degree in Electrical Engineering from Chang Gung University in 2014. His research interests include information security, visual cryptography and RFID security.

Erl-Huei Lu received the B.S. and M.S. degrees in electrical engineering from Chung Cheng Institute of Technology, Taiwan, in 1974 and 1980, respectively, and the Ph.D. degree electrical engineering from National Cheng Kung University, Taiwan, in 1988. Lu is a professor in the Department of Electrical Engineering, Chang Gung University, Taiwan. His research interests include error-control coding, network security, and systolic architectures.

Guide for Authors
International Journal of Network Security

IJNS will be committed to the timely publication of very high-quality, peer-reviewed, original papers that advance the state-of-the art and applications of network security. Topics will include, but not be limited to, the following: Biometric Security, Communications and Networks Security, Cryptography, Database Security, Electronic Commerce Security, Multimedia Security, System Security, etc.

1. Submission Procedure

Authors are strongly encouraged to submit their papers electronically by using online manuscript submission at http://ijns.jalaxy.com.tw/.

2. General

Articles must be written in good English. Submission of an article implies that the work described has not been published previously, that it is not under consideration for publication elsewhere. It will not be published elsewhere in the same form, in English or in any other language, without the written consent of the Publisher.

2.1 Length Limitation:

All papers should be concisely written and be no longer than 30 double-spaced pages (12-point font, approximately 26 lines/page) including figures.

2.2 Title page

The title page should contain the article title, author(s) names and affiliations, address, an abstract not exceeding 100 words, and a list of three to five keywords.

2.3 Corresponding author

Clearly indicate who is willing to handle correspondence at all stages of refereeing and publication. Ensure that telephone and fax numbers (with country and area code) are provided in addition to the e-mail address and the complete postal address.

2.4 References

References should be listed alphabetically, in the same way as follows:

For a paper in a journal: M. S. Hwang, C. C. Chang, and K. F. Hwang, ``An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445--446, 2002.

For a book: Dorothy E. R. Denning, *Cryptography and Data Security*. Massachusetts: Addison-Wesley, 1982.

For a paper in a proceeding: M. S. Hwang, C. C. Lee, and Y. L. Tang, ``Two simple batch verifying multiple digital signatures," in *The Third International Conference on Information and Communication Security (ICICS2001)*, pp. 13--16, Xian, China, 2001.

In text, references should be indicated by [number].

Subscription Information

Individual subscriptions to IJNS are available at the annual rate of US$ 200.00 or NT 7,000 (Taiwan). The rate is US$1000.00 or NT 30,000 (Taiwan) for institutional subscriptions. Price includes surface postage, packing and handling charges worldwide. Please make your payment payable to "Jalaxy Technique Co., LTD." For detailed information, please refer to http://ijns.jalaxy.com.tw or Email to ijns.publishing@gmail.com.