IJNS
International Journal of Network Security

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

Vol. 22, No. 3 (May 2020)

INTERNATIONAL JOURNAL OF NETWORK SECURITY

Editor-in-Chief

Prof. Min-Shiang Hwang
Department of Computer Science & Information Engineering, Asia University, Taiwan

Co-Editor-in-Chief:

Prof. Chin-Chen Chang (IEEE Fellow)
Department of Information Engineering and Computer Science, Feng Chia University, Taiwan

Publishing Editors

Shu-Fen Chiou, Chia-Chun Wu, Cheng-Yi Yang

Board of Editors

Ajith Abraham
School of Computer Science and Engineering, Chung-Ang University (Korea)

Wael Adi
Institute for Computer and Communication Network Engineering, Technical University of Braunschweig (Germany)

Sheikh Iqbal Ahamed
Department of Math., Stat. and Computer Sc. Marquette University, Milwaukee (USA)

Vijay Atluri
MSIS Department Research Director, CIMIC Rutgers University (USA)

Mauro Barni
Dipartimento di Ingegneria dell'Informazione, Università di Siena (Italy)

Andrew Blyth
Information Security Research Group, School of Computing, University of Glamorgan (UK)

Chi-Shiang Chan
Department of Applied Informatics & Multimedia, Asia University (Taiwan)

Chen-Yang Cheng
National Taipei University of Technology (Taiwan)

Soon Ae Chun
College of Staten Island, City University of New York, Staten Island, NY (USA)

Stefanos Gritzalis
University of the Aegean (Greece)

Lakhmi Jain
School of Electrical and Information Engineering, University of South Australia (Australia)

Chin-Tser Huang
Dept. of Computer Science & Engr, Univ of South Carolina (USA)

James B D Joshi
Dept. of Information Science and Telecommunications, University of Pittsburgh (USA)

Ç etin Kaya Koç
School of EECS, Oregon State University (USA)

Shahram Latifi
Department of Electrical and Computer Engineering, University of Nevada, Las Vegas (USA)

Cheng-Chi Lee
Department of Library and Information Science, Fu Jen Catholic University (Taiwan)

Chun-Ta Li
Department of Information Management, Tainan University of Technology (Taiwan)

Iuon-Chang Lin
Department of Management of Information Systems, National Chung Hsing University (Taiwan)

John C.S. Lui
Department of Computer Science & Engineering, Chinese University of Hong Kong (Hong Kong)

Kia Makki
Telecommunications and Information Technology Institute, College of Engineering, Florida International University (USA)

Gregorio Martinez
University of Murcia (UMU) (Spain)

Sabah M.A. Mohammed
Department of Computer Science, Lakehead University (Canada)

Lakshmi Narasimhan
School of Electrical Engineering and Computer Science, University of Newcastle (Australia)

Khaled E. A. Negm
Etisalat University College (United Arab Emirates)

Joon S. Park
School of Information Studies, Syracuse University (USA)

Antonio Pescapè
University of Napoli "Federico II" (Italy)

Chuan Qin
University of Shanghai for Science and Technology (China)

Yanli Ren
School of Commun. & Infor. Engineering, Shanghai University (China)

Mukesh Singhal
Department of Computer Science, University of Kentucky (USA)

Tony Thomas
School of Computer Engineering, Nanyang Technological University (Singapore)

Mohsen Toorani
Department of Informatics, University of Bergen (Norway)

Sherali Zeadally
Department of Computer Science and Information Technology, University of the District of Columbia, USA

Jianping Zeng
School of Computer Science, Fudan University (China)

Justin Zhan
School of Information Technology & Engineering, University of Ottawa (Canada)

Ming Zhao
School of Computer Science, Yangtze University (China)

Mingwu Zhang
College of Information, South China Agric University (China)

Yan Zhang
Wireless Communications Laboratory, NICT (Singapore)

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

© Jalaxy Technology Co., Ltd., Taiwan 2005
23-75, P.O. Box, Taichung, Taiwan 40199, R.O.C.

Volume: 22, No: 3 (May 1, 2020)

International Journal of Network Security

A Taxonomy of User Authentication Schemes for Multi-server Environments

Hung-Wei Yang[1], Hsieh-Tsen Pan[1], Yung-Hsing Chen[2], and Min-Shiang Hwang[1,3]
(Corresponding author: Min-Shiang Hwang)
Department of Computer Science & Information Engineering, Asia University, Taichung, Taiwan[1]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taichung, Taiwan, R.O.C.
Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan[2]
Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan[3]
(Email: mshwang@asia.edu.tw)
(Received Mar. 21, 2020; Revised and Accepted Apr. 11, 2020; First Online Apr. 12, 2020)

Abstract

In the conventional scheme, the user authentication scheme is only considered for a single specific server. However, in many distributed environments, resources and services are distributed across multiple servers. If the conventional user authentication scheme is applied to a multi-server environment, each server should manage its own user authentication process, and users must register with each server and maintain a separate login ID and password pair to access each server. Therefore, applying the conventional user authentication scheme to a multi-server environment is inefficient and impractical. In this article, we propose to establish a taxonomy of user authentication schemes for multi-server environments.

Keywords: Multi-server; Password; Smart Card; User Authentication

1 Introduction

In the era of information explosion, the Internet has become a part of our lives. With the rapid development of the Internet, users can obtain various services from the servers through the Internet. However, transmitting messages between users and servers over an unsecured network may be subject to certain attacks [13]. Using password-based user authentication with smart cards is one of the effective solutions to protect these services. In recent decades, many schemes using passwords and smart cards to authenticate users and remote servers have been proposed [14, 38].

In the conventional scheme, the user authentication scheme is only considered for a single specific server. However, in many distributed environments, resources and services are distributed across multiple servers, and these servers work together to provide services to their users. If the conventional user authentication scheme is applied to a multi-server environment, each server should manage its own authentication process, and the user must register with each server separately and maintain a separate login identity and password pair to access various servers. Therefore, applying the conventional user authentication scheme to a multi-server environment is inefficient and impractical. With the development of multi-server systems, the user authentication schemes used in multi-server environments have attracted more and more attention in recent decades [2–5, 8, 10, 11, 15, 16, 22, 23, 26, 27, 30, 32, 33, 35, 39, 41, 45, 46].

The rest of the article is organized as follows. Section 2 introduces four topologies for user authentication schemes for multiple servers. Section 3 presents some requirements for user authentication schemes designed for multiple servers. We will classify each category proposed by the multi-server in Section 4. Finally, Section 5 summarizes this article and points out future research prospects.

2 Topologies of User Authentication Schemes for Multi-server

In a multi-server environment, there are four topologies for verifying user identity.

1) Traditional user authentication schemes for multi-server environments.

   Apply the existing user authentication scheme for a single server environment directly to multiple server environments (see Figure 1). This method is simple and easy. However, with this method, users must remember multiple sets of usernames and passwords for all registered servers. Users want to get some services from different remote servers. He / she needs to log in to different servers with different passwords. The user must register the different service servers provided, and then the user must remember many

pairs of IDs and passwords.

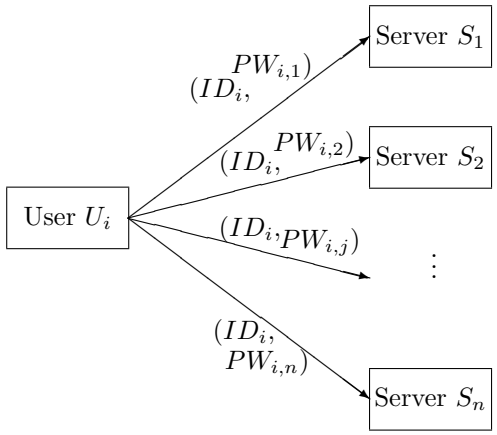


Figure 1: Traditional user authentication schemes for multi-server environments.

2) Proxy-based schemes for multi-server environment.

The user logs into one of multiple servers through a third-party agent (see Figure 2). In this method, the user only remembers a set of usernames and passwords of the proxy server. Another set of usernames and passwords for all multi-servers are kept in the proxy server. The main advantage of this method is that users only remember a set of usernames and passwords. However, all usernames and passwords are reserved by the proxy server. If the proxy server is attacked, all users in the proxy server will be disguised as illegal users.

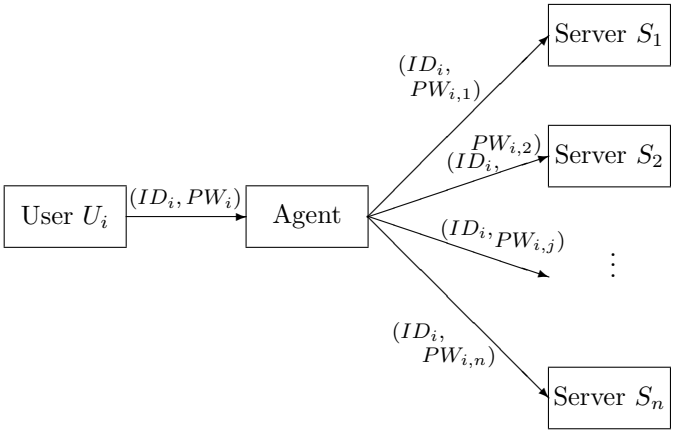


Figure 2: Proxy-based schemes for multi-server environment

3) Ticket-based schemes for multi-server environment.

The user first registers on the proxy server. Next, the proxy server registers with other servers to provide services to legitimate users. The proxy server will obtain tickets from these servers for future login to these servers and distribute them to users (see Figure 3). The main advantage of this method is that the proxy server does not have to be a trusted server. Because the proxy server just logs in to the server for the user. The login message is for a one-time ticket, and the user can log in to the server at one time. However, the main disadvantage of this method is that all users need to keep all tickets for all servers.

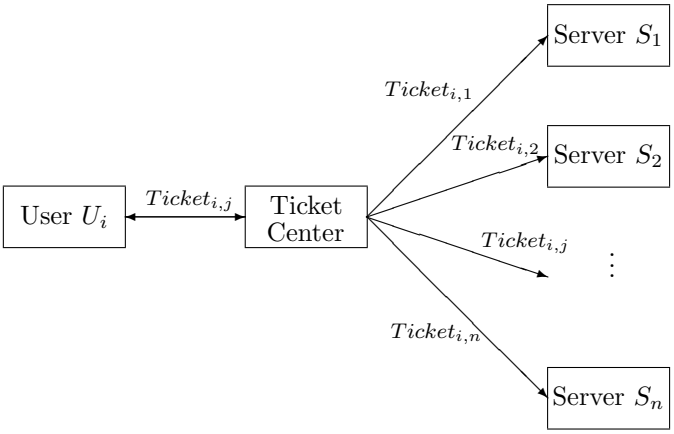


Figure 3: Ticket-based schemes for multi-server environment

4) Smart Card-based schemes for multi-server environment.

Each user has his/her own smart card, which stores all the identity and password pairs that all servers provide services. The user remembers only one set of username and password. However, these username and password sets are different for all servers. Therefore, one server cannot be disguised as a legitimate user on another server (see Figure 4). The main disadvantage is that it requires additional equipment, namely a smart card reader. In addition, smart cards must be tamper-proof and tip-proof devices. This means that if the smart card is stolen or lost, the scheme can resist various attacks.

3 Requirement

The user authentication scheme used in a multi-server environment should meet the following basic functional criteria [16, 26, 43]:

F1: Single Registration. The user only needs to register once in the registration center, and then he/she can access all the servers in the system. It is consistent with the multi-server network architecture and does not require repeated registration.

F2: No Verification Table. The registration center or server does not have to maintain any verification or password tables used to verify the identity of users.

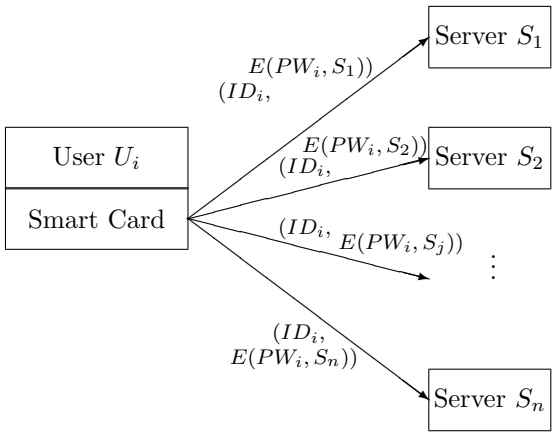


Figure 4: Smart Card-based schemes for multi-server environment

F3: User Friendly. The user can freely choose and change his/her password at any time without any help from the registration center.

F4: Efficiency. The computational costs in each stage (ie, user registration, server registration, login, authentication, session key establishment, and password change stage) should be as efficient as possible. The memory used in the server and smart card, and the communication between the user and the server should also be as small as possible.

F5: Mutual Authentication. Users and servers can authenticate each other. This means that the server can not only authenticate legitimate users, but also users can authenticate legitimate servers. Similarly, if necessary, the server and the registration center can also authenticate each other.

F6: Session Key Agreement. Users and servers can establish session keys with each other to protect their subsequent confidential communications. Apart from the user and the corresponding server, no one knows about the agreed session key.

F7: User Anonymity. User anonymity requires that only the server knows the identity of the user with whom he / she is interacting, and no third party can know the identity of the user. Even if the attacker intercepts the communication message between the legitimate user and the server, the attacker cannot obtain the identity of the user.

In addition to the above basic functional criteria, the following security criteria should be further checked [15, 16, 39, 43]:

S1: Perfect Forward Secrecy. Even if the long-term secret key is leaked, the established session key should not be derived. Therefore, the previous confidential communications can be protected.

S2: Known-key Security. No one can use the known secret key or secret information of the legitimate user to obtain the session key between the user and the server.

S3: Resist Gussing Attacks. The entropy of most passwords is too low to be vulnerable to password guessing attacks. There are two types of guessing attacks: online guessing attacks and offline guessing attacks. In an online guessing attack, the attack guesses the password of a legitimate user, and then attempts to use the password to log in to a server on the Internet. In an offline guess attack, the attacker intercepted certain messages transmitted between the user and the server. Next, the attacker attempts to guess the password and check it through the intercepted message.

S4: Resist Replay Attacks. If an attacker eavesdrops on a previous login message sent by a legitimate user and attempts to replay the same message to pretend that the user passed the authentication process, this attempt is called a replay attack. User authentication schemes should resist replay attacks.

S5: Resist Attacks from Stolen Verifiers. If some important information stored in the authentication center (server) is stolen, the attacker will use this information to disguise a legitimate user in order to obtain services from the server. To address this attack, the system is designed not to maintain any verification tables or password tables.

S6: Resist Server Spoofing Attacks. A server spoofing attack refers to an attacker posing as another server on the Internet to obtain a user's private message (such as identity and password). Once an attacker pretends to be a server (registration center or authentication server) to deceive and obtain certain private messages of a legitimate user or registration center, the attacker can successfully pretend to be the legitimate user or registration center.

S7: Resist Impersonation Attacks. An impersonation attack refers to an attacker who pretends to be a legitimate user. Its purpose is to deceive the server or the registration center on the Internet to obtain certain services from the server. Once the attacker pretends to be successfully authenticated as a legitimate user, the attacker can use all services in the server.

S8: Resist Man-in-the-middle Attacks. A man-in-the-middle attack refers to an attacker who is secretly hidden between the user and the server. Its purpose is to obtain some important messages transmitted between the user and the server, these users believe that they are in direct communication.

S9: Resist Insider Attacks. An insider attack refers to an attacker who has been authorized to access

a computer system and may also have confidential information or important knowledge of the system. Usually, the attacker is a system administrator with authorized system access. Insiders who carry out attacks have obvious advantages over external attackers because they have authorized system access rights and may be familiar with system architecture.

S10: Resist Outsider Attacks. The outsider is someone who has been registered on the server, not someone who is not a system user. The attacker is one of the legitimate users of the server and owns messages sent by the server to log in to the server. Attackers can use their private messages and certain messages intercepted between the victim legal user and the server, to obtain some confidential messages of the victim legal user, disguise the user, or obtain some server services.

S11: Resist Parallel Session Attacks. Without knowing the user's password, an attacker can eavesdrop on the communication between the user and the server to create a valid login message, thereby pretending to be a legitimate user.

S12: Resist Smart Card Loss Attacks. When a smart card is lost or stolen, an unauthorized user can easily change the smart card's password, or can guess the user's password by using a password guessing attack, or can impersonate the user to log in to the system.

S13: Resist Denial of Service Attacks. The attacker destroyed the authentication information between the user and the server, so the legitimate user can no longer successfully log in, or the server no longer performs authentication within the valid time.

4 Types of User Authentication Schemes for Multi-server

Generally speaking, user authentication schemes designed for multi-server systems involve three types of parties: registration centers, users, and a set of servers. If a user wants to use the services or resources provided by the servers belonging to the multi-server system, he/she only needs to register once in the registration center instead of registering on each server separately. After registration, access to the user will be granted and the user will be allowed to log in to any server in the system.

In the user authentication scheme, there are three participants: login users, various servers, and registration centers (RC) or system administrators (SA). Generally, the user authentication schemes for multiple servers can be divided into six phases:

1) Initialization Phase: Before the system, SA needs to set up and publish some information for registered users.
2) Registration Phase: Before logging into the server, new users must first register some information to become legal users. The registration phase is only performed once. After completing this phase, each legitimate user will get a valid user ID and password.
3) Login Phase: The user types his/her identity and password to log in to any server.
4) Authentication (Verification) Phase: The server verifies the legitimacy of the remote login user.
5) Mutual Authentication and Session Key Agreement Phase: At this phase, if the authentication is successful, the server and user will authenticate each other and agree on a common session key to ensure the security of further communication.
6) Password Change Phase: By performing this phase, the user can change his/her password as he/she wishes, without connecting to the registration center or any server.

We classify the user authentication scheme for multi-server environment as follows:

1) Based on the neural network [18, 23, 27, 28];
2) Based on the geometry property [5, 5, 15, 16, 19, 29];
3) Based on the Diffie-Hellman key agreement [6, 7, 29];
4) Based on the dynamic identity [1, 6, 12, 21, 24–26, 31, 34, 36, 44];
5) Based on the QR [6, 7, 9, 29, 37, 43];
6) Based on one-way hash function [39, 43].

4.1 The NN-based Authentication Scheme for Multi-Server

In the neural network-based authentication scheme for multi-server, each legitimate user has only one user identity and its corresponding password. System administrators (SA) identify users through neural networks. Train user identity and password pairs in the neural network, and store the weights in each server.

In 2001, Li et al. [23] first proposed a remote password authentication scheme for multi-server architecture using neural network. Users only need to register once in the registration center, and remember the ID and password pairs, users can get the services provided.

On the other hand, Ku et al. [18] pointed out in 2005 that Li et al.'s scheme [23] scheme could not resist password guessing attacks and internal attacks, so Lin proposed an improved scheme to eliminate these vulnerabilities in 2008 [27]. However, the high cost of using neural networks in their solutions usually requires a lot of time to build, train, and maintain neural networks [23, 27].

4.2 The Geometry-based Authentication Scheme for Multi-Server

In 2005, Chang and Kuo proposed another user authentication scheme with key agreement based on Chinese residual theorem and modulus table [5]. Hwang and Shiau proved the lack of explicit key authentication and communication cost efficiency in Juang's and Chang-Kuo's schemes [5, 16]. Therefore, they developed an improved authentication key agreement based on geometric scheme. However, forward confidentiality is still lacking, and each server must maintain a user table in accordance with Huang and Shiau's scheme.

In 2003, Lin et al. proposed a user authentication scheme for multi-server architecture based on the geometric characteristics of the Euclidean scheme [29]. In their scheme, Lin et al. assumes that the SA is trustworthy. SA sets several public and secret parameters. Every legitimate user can get the granted services from his/her registered server. Users do not need to register repeatedly on various servers. Unfortunately,Ku et al. [19] and Cao et al. [3] proves that Lin et al.'s scheme has been subjected to forgery attacks and password guessing attacks and cannot be repaired at all.

4.3 The Hash-based Authentication Scheme for Multi-Server

Tsai proposed a multi-server authentication scheme based on one-way hash function in 2008 [39]. Tsai's scheme is effective because it only involves lightweight operations such as hashing and XOR. However, his scheme is not secure. Wang et al. showed that Tsai's user authentication scheme cannot resist server spoofing and impersonation attacks in 2009 [43]. Therefore, they proposed an improved user authentication scheme to resist these weaknesses. Unfortunately, there are still two vulnerabilities in the Tsai and Wang et al.'s schemes. First, the session key established between the user and the server does not satisfy perfect forward secrecy. Once the master key is leaked, all the session keys used can be derived. Therefore, the confidentiality between the user and the server is no longer retained. Second, the registration center knows all the session keys established between the user and the server. Therefore, the confidentiality of the registration center is considered to be risky. Therefore, in order to provide completely confidential communication between the user and the server, these two existing problems should be considered more carefully.

In 2013, Chen et al. [7] also discovered some flaws in Tsai's scheme [39]. The session key established by Tsai's and Wang et al.'s schemes [39, 43] lacks forward secrecy and is known by RC. On the other hand, due to the lack of forward secrecy, it will allow attackers to derive the agreed session key and decrypt all transmissions while losing the master key. In addition, since the session key is disclosed to the RC, the confidentiality of the transmission will become a challenge. Although the RC is generally regarded as a trusted party, the fact that the RC has knowledge of the session key always makes the user worry about the confidentiality of the communication. To eliminate doubt, the scheme that prevents RC from knowing the session key is more persuasive and preferable. As a result, Tsai's scheme cannot resist camouflage attacks and server spoofing attacks. Wang's work also pointed out the type of attack [43].

4.4 The QR-based Authentication Scheme for Multi-Server

In 2009, Wang et al. proposed a multi-server user authentication scheme with key agreement using smart cards and quadratic residue (QR) [43]. In 2013, Chen et al. proved two weaknesses in Wang et al's scheme, such as the lack of forward confidentiality and insufficient session key protection for RC [7].

In 2011, Tan also found that Wang et al.'s scheme could not withstand impersonation attacks and could not provide perfect forward security [37]. Tan proposed an improved scheme based on the Diffie-Hellman assumption. However, Feng et al. found that Tan's scheme could not resist password guessing attacks [9].

4.5 The ID-based Authentication Schemes for Multi-Server

In 2009, Liao and Wang proposed a remote user authentication scheme based on dynamic ID for multi-server environments [26]. The user anonymity by replacing static IDs with dynamic IDs makes their scheme more suitable for specific applications such as e-commerce and online shopping services. Unfortunately, Chen et al. showed that if there are malicious insiders, Liao and Wang's scheme is vulnerable to impersonation attacks, and the agreed session key between the user and the server is not secure [6]. Anyone who obtains the secret keys $h(x)$ and y can not only derive all the session keys between the user and the server, but also can pretend that any user can successfully log in to any server.

In addition, Hsiang and Shih also pointed out that Liao et al.'s scheme [26] is still vulnerable to attacks from insiders, masquerade, and server spoofing attacks, so the scheme cannot achieve mutual authentication [12]. Therefore, Hsiang and Shih proposed an improved scheme to Liao et al.'s scheme. However, their scheme is still vulnerable to masquerade attacks and smart card loss attacks [12]. In addition, their scheme cannot provide mutual authentication.

But in 2011, Sood et al. [36] and Lee et al. [21] proved that Hsiang et al.'s scheme [12] was still insecure. Sood et al. proposed an improved scheme, which authenticates the user's identity through the registration center [36]. Li et al. [25] and Xue et al. [44] continue to conduct verification research in the registration center. Instead, Lee et al.'s scheme [21] relies on the service server to verify the user's identity. Li et al. [24] remedied Lee et al.'s scheme

in 2013 to prevent forgery attacks, server spoofing attacks, and easily change passwords. However, Ling et al. proved that Li et al.'s scheme was vulnerable to server spoofing attack in 2015 [31].

In 2016, Amin proposed a remote user authentication scheme for a multi-server environment with smart cards [1]. But Pan et al. said that Amin's remote user authentication scheme is vulnerable to offline identity guessing by smart card theft attacks and offline password guessing by smart card theft attacks [34].

4.6 The DH-based Authentiovation Scheme for Multi-Server

In 2001, Tsaur proposed a flexible user authentication scheme for multi-server services [40]. Unfortunately, Kim et al. showed that Tsaur's scheme cannot defend against password guessing attacks [17]. Since then, Tsaur et al. proposed an improved scheme using RSA cryptosystem and Lagrange interpolation method in 2005 [42]. However, their scheme is inefficient in terms of computation and communication because it takes a lot of time and information to construct the interpolation polynomial.

Juang proposed a user authentication scheme with authenticated key agreement for a multi-server environment using a symmetric cryptosystem in 2004 [16]. In his scheme, the functions of mutual authentication and session key agreement are provided. Nonetheless, Ku et al. proves that Juang's scheme cannot resist insider attacks and does not provide perfect forward secrets [20].

In 2011, Tan discovered that Wang et al.'s scheme [43] could not withstand impersonation attacks and could not provide perfect forward security [37]. Tan proposed an improved scheme based on the Diffie-Hellman assumption [13]. However, Feng et al. found that Tan 's plan could not refuse the password guess the attack [9].

In 2013, Chen et al. [7] proposed a scheme that not only meets all security requirements, but also resists all well-known attacks, and can make up for the weaknesses of the Tsai's scheme [39] and Wang et al.'s scheme [43].

5 Conclusion

In this article, we put forward some requirements and criteria for designing user authentication schemes in a multi-server environment. We also introduced the topology used for multi-server environments and the types of user authentication schemes.

Acknowledgments

This research was partially supported by the Ministry of Science and Technology, Taiwan (ROC), under contract no.: MOST 108-2410-H-468-023 and MOST 108-2622-8-468-001-TM1.

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] S. Banerjee, M. P. Dutta, and C. T. Bhunia, "An improved smart card based anonymous multi-server remote user authentication scheme," *International Journal of Smart Home*, vol. 9, no. 5, pp. 11–22, 2015.

[3] X. Cao and S. Zhong, "Breaking a remote user authentication scheme for multi-server architecture," *IEEE Communications Letters*, vol. 10, no. 8, pp. 11–22, 2006.

[4] P. Chandrakar and H. Om, "A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC," *Computer Communications*, vol. 110, pp. 26–34, 2017.

[5] C. C. Chang and J. Y. Kuo, "An efficient multi-server password authenticated key agreement scheme using smart cards with access control," in *19th IEEE International Conference on Advanced Information Networking and Applications (AINA'05)*, vol. 2, pp. 257–260, Taiwan, Mar. 2005.

[6] T. Y. Chen, C. C. Lee, M. S. Hwang, and J. K. Jan, "Cryptanalysis of a secure dynamic ID based remote user authentication scheme for multi-server environment," in *Fourth International Conference on Innovative Computing, Information and Control (ICICIC'09)*, pp. 725–728, 2009.

[7] T. Y. Chen, C. C. Lee, M. S. Hwang, and 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, 2013.

[8] A. Choubey and K. Chatterjee, "Secure remote user authenticationformulti-server environmentusing machine learning technique," in *International Conference on Circuit, Power and Computing Technologies (ICCPCT'16)*, pp. 1–5, Bihar, India, 2016.

[9] 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.

[10] S. Gaharana and D. Anand, "Dynamic ID based remote user authentication in multi server environment using smart cards: A review," in *International Conference on Computational Intelligence and Communication Networks*, pp. 1081–1084, 2015.

[11] M. Guan, J. Song, and W. Liu, "A threshold multi-server protocol for password-based authentication," in *3rd International Conference on Cyber Security and Cloud Computing*, pp. 108–118, 2016.

[12] H. C. Hsiang and W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication

scheme for multi-server environment," *Computer Standards and Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.

[13] M. S. Hwang and I. C. Lin, *Introduction to Information and Network Security (6ed, in Chinese)*. Taiwan: Mc Graw Hill, 2017.

[14] 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, Jan. 2010.

[15] R. J. Hwang and S. H. Shiau, "Provably efficient authenticated key agreement protocol for multi-servers," *The Computer Journal*, vol. 50, no. 5, pp. 602–615, 2007.

[16] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 251–255, November 2004.

[17] S. Kim, S. Lim, and D. Won, "Cryptanalysis of flexible remote password authentication scheme of ICN01," *Electronics Letters*, vol. 38, no. 24, pp. 1519–1520, 2002.

[18] W. C. Ku, "Weaknesses and drawbacks of a password authentication scheme using neural networks for multiserver architecture," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 1002–1005, 2005.

[19] W. C. Ku, S. T. Chang, and M. H. Chiang, "Weaknesses of a remote user authentication scheme using smart cards for multi-server architecture," *IEICE Transactions on Communications*, vol. E88-B, no. 8, pp. 3451–3454, 2005.

[20] W. C. Ku, H. M. Chuang, and M. H. Chiang, "Cryptanalysis of a multi-server password authenticated key agreement scheme using smart cards," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, pp. 3235–3238, Nov. 2005.

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

[22] C. T. Li, C. C. Lee, C. Y. Weng, and C. I Fan, "An extended multi-server-based user authentication and key agreement scheme with user anonymity," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 1, p. 119–131, 2013.

[23] L. H. Li, I. C. Lin, and M. S. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.

[24] X. Li, J. Ma, W.D. Wang, Y. P. Xiong, and J. S. Zhang, "A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments," *Mathematical and Computer Modelling*, vol. 58, pp. 85–95, 2013.

[25] X. Li, Y. P. Xiong, J. Ma, and W. D. Wang, "An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards," *Journal of Network and Computer Applications*, vol. 35, no. 2, p. 763–769, 2012.

[26] Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards and Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.

[27] I. C. Lin, "A neural network system for authenticating remote users in multi-server architecture," *International Journal of Communication Systems*, vol. 21, pp. 435–445, 2008.

[28] I. C. Lin, H. H. Ou, and M. S. Hwang, "A user authentication system using back-propagation network," *Neural Computing & Applications*, vol. 14, no. 3, pp. 243–249, 2005.

[29] I. C. Lin, M. S. Hwang, and 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.

[30] Y. Lin, K. Wang, B. Zhang, Y. Liu, and X. Li, "An enhanced biometric-based three factors user authentication scheme for multi-server environments," *International Journal of Security and Its Applications*, vol. 10, no. 1, pp. 315–328, 2016.

[31] C. H. Ling, W. Y. Chao, S. M. Chen, and M. S. Hwang, "Cryptanalysis of dynamic identity based on a remote user authentication scheme for a multi-server environment," in *Advances in Engineering Research, vol. 15*, pp. 981–986, 2015.

[32] N. M. R. Lwamo, L. Zhu, C. Xu, K. Sharif, X. Liu, and C. Zhang, "Suaa: A secure user authentication scheme with anonymity for the single & multi-server environments," *Information Sciences*, vol. 477, pp. 369–385, 2019.

[33] T. Maitra, S. K. H. Islam, R. Amin, D. Giri, M. K. Khan, and N. Kumar, "An enhanced multi-server authentication protocol using password and smart-card: cryptanalysis and design," *Security and Communication Networks*, vol. 9, pp. 4615–4638, 2016.

[34] H. T. Pan, C. S. Pan, S. C. Tsaur, and M. S. Hwang, "Cryptanalysis of efficient dynamic ID based remote user authentication scheme in multi-server environment using smart card," in *12th International Conference on Computational Intelligence and Security (CIS'16)*, pp. 590–593, 2016.

[35] S. S. Sahoo, S. Mohanty, and M. Polai, "A secure biometric based user authentication scheme for multi-server environment using chaotic map," in *6th International Conference on Signal Processing and Integrated Networks (SPIN'19)*, pp. 637–642, 2019.

[36] S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609–618, 2011.

[37] Z. Tan, "Improvement on a password authentication scheme for multi-server environments," *Journal of Convergence Information Technology*, vol. 6, no. 1, pp. 218–228, 2011.

[38] 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.
[39] J. L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Computers and security*, vol. 27, pp. 115–121, 2008.
[40] W. J. Tsaur, "A flexible user authentication scheme for multi-server internet services," Lecture Notes in Computer Science, vol. 2093, pp. 174–183, Springer-Verlag, 2001.
[41] W. J. Tsaur, C. C. Wu, and W. B. Lee, "A smart card-based remote scheme for password authentication in multi-server internet services," *Computer Standards and Interfaces*, vol. 27, pp. 39–51, 2004.
[42] W. J. Tsaur, C. C. Wu, and W. B. Lee, "An enhanced user authentication scheme for multi-server internet services," *Applied Mathematics and Computation*, vol. 170, pp. 258–266, 2005.
[43] R. C. Wang, W. S. Juang, and C.L. Lei, "User authentication scheme with privacy-preservation for multi-server environment," *IEEE Communications Letters*, vol. 13, no. 2, pp. 157–159, 2009.
[44] K. P. Xue, P. L. Hong, and C. S. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences*, vol. 80, pp. 195–206, 2014.
[45] B. Ying and A. Nayak, "Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography," *Journal of Network and Computer Applications*, vol. 131, pp. 66–74, 2019.
[46] J. Zhang, J. Ma, X. Li, and W. Wang, "A secure and efficient remote user authentication scheme for multi-server environments using ECC," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 8, pp. 2930–2947, 2014.

Biography

Hung-Wei Yang received B.S. in Industry Engineer From Da-Yeh University, Taiwan in 2001; M.S. in Information Management, Chao Yang University, Taiwan in 2009; Doctoral Program of Information Engineering, Asia University, Taiwan from 2016 till now. From 2012 to 2014, he was the manager in International Business Machine. From 2014 to 2015, he was the manager in Cisco Systems, Inc. Taiwan branch. From 2016 to 2019 he is the sales director of China branch in Syntron Technology Co. Ltd. Taipei Taiwan .From 2020 he is channel director in M-Power Co. Ltd., Taipei Taiwan.

Hsien-Tsen Pan received B.S. in Business Administration From Soochow University, Taiwan in 1999; M.S. in Information Engineering, Asia University, Taiwan in 2015; Doctoral Program of Information Engineering, Asia University, Taiwan from 2015 till now. From 2011 to 2014, he was the manager in Enterprise Service Chunghwa Telecom South Branch Taichung Taiwan. From 2014 to 2017, he was the operation manager in Medium division Taiwan Ricoh Co., Ltd. Taichung Taiwan From 2017 Sep 20 he is the Apple MDM Server Service VP in Get Technology Co.Ltd. Taipei Taiwan.

Yung-Hsing Chen received the B.S. in Management Information Systems from National Chung Hsing University, Taiwan in 2019. Currently, he is a research assistant at the department of Computer Science and Information Engineering, Asian University, Taiwan. His current research interests include information security, blockchain, Internet of Things, artificial intelligence, e-commerce, and intellectual property law.

Min-Shiang Hwang received M.S. in industrial engineering from National Tsing Hua University, Taiwan in 1988; and Ph.D. degree in computer and information science from National Chiao Tung University, Taiwan in 1995. 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 Department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Exceilent 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 Department of Computer Science and Information Engineering, AU. His current research interests include information security, electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.

Role Mining Algorithms Satisfied the Permission Cardinality Constraint

Jingyu Wang, Jingnan Dong, and Yuesheng Tan
(Corresponding author: Jingnan Dong)

School of Information Engineering, Inner Mongolia University of Science and Technology
7 Aerding Street, Baotou City, Inner Mongolia, 014010, People's Republic of China
(Email: 867324154@qq.com)

(Received Nov. 16, 2018; Revised and Accepted Mar. 17, 2019; First Online June 16, 2019)

Abstract

With the increase of access control size in big data, the roles in most of the existing role mining algorithms have overmuch permission, which conveniently results in fraud. Therefore, this paper proposes two kinds of role mining algorithms that satisfy the permission cardinality constraints. Algorithms 1 divide each row of the sorted access control matrix to generate a set of roles. Algorithm 2 intersect the permission of adjacent users of the ordered access control matrix to generate a candidate role sets. The iterative reduction is then performed multiple times on the basis of the candidate role sets to produce most of the decisive role sets. Both algorithms first perform row and column sorting on the access control matrix, which draw on word frequency statistics and frequent item mining, respectively. The data applied in the experiment are a common real data set. Relevant experimental results demonstrate that the performance of algorithm 2 is superior to algorithm 1 and PRUCC-RM.

Keywords: Access Control; Frequent Item Mining; Permission Cardinality Constraints; Role Mining; Word Frequency

1 Introduction

Role based access control has been one of the most widespread way of access control.The key of applying the model is the definition of the roles. The solution of the question is divided in two types, one is a top-down method, which gain roles by analyzing users' circumstance and the business process [18], the other way is a down-top way,which utilize data mining technology to find roles from existing user permission assignment relationship (UPA). The user permission assignment relation that specifies which individuals had access to which resources in the original system can be presented in the form of a boolean matrix. Mitra divides the role mining model into the deterministic model and the probabilistic model [12]. Vaidya defined the basic role mining problem that finding a minimal set of role from an input UPA that provides an equivalent user permission assignment [19], who shows the problem of finding the minimal set of descriptive roles and relationships without disturbing permission assignments is NP-complete. There exist various ways to mine roles.Belim presents an algorithm for analyzing the matrix of authorized user permission for optimal role formation [1], which solves the role mining problem with the help of graph theory knowledge.Dong uses bipartite network models for role mining and solves problems from the perspective of edge importance in complex networks [3]. Huang converted Basic role mining problem to the Set Cover Problem [6]. There are a few role mining algorithms with machine learning model. Constraints are a powerful mechanism for arranging high-level organizational strategies. In addition, there are malicious activities in RBAC [15], by applying the constraints, the rate in database attacks can be reduced, and fraudulent behavior can be prevented. It is necessary for enterprises to conduct role mining meeting constraints in implementing RBAC. Ye proposed a novel role mining approach useing answer set programming (ASP) that meets various optimization objectives, named constrained role miner [21].

There are three constraints in role-based access control. That is separation of duties, cardinality constraints and prerequisite constraints.Separation of duties is widely used in situations where multiple people need to work together to perform a sensitive task, but not by fewer people to prevent fraud. There are for two situations about prerequisite constraint. One is prerequisite constraint on the role which specifies that one user can obtain role $r1$ only after the user has obtained role $r2$. The other is prerequisite constraint of permission which specifies that a permission can be assigned to a certain role only after the role possesses permission p. There are four situations about cardinality constraint, one is permission cardinality which specifies the maximum number of permission a role can have in a RBAC system; Second is user cardinality constraint which specifies the maximum number of user a role can belong to; Third is role usage cardinality con-

straint which specifies the limited number of roles which each user can have; The last one is permission assignment cardinality constraint which specifies the limited number of roles that each permission can be assigned.

The rest of the paper is framed as follows: Section 2 reflects the related work. Section 3 exhibits the related terms needed to understand the algorithms. Section 4 introduces in detail the execution process and flow chart of the two algorithms proposed in this paper, and carries on the algorithm analysis. Section 5 uses real public data for algorithm experiments,and analyses the results. Finally, we conclude our work in Section 6.

2 Related Works

Although the algorithms of role mining have been proposed [7], the role which most algorithm generate don't meet cardinality constraint [16], which makes many roles own overmuch permission, violating separation of duty constraints easily, and being harmful to redistribution. For roles generated meet permission cardinality constraints. One which uses the combination of clustering and limited permission set mining gets roles with meeting given cardinality constraint [8], it is very time consuming to mine role from a collection of disorganized user permission, because a role is mined each time. The statistics of the user's no-visited permission are required, and also calculates all possible intersections of all users' no-visited permission. Role usage cardinality constraint and permission assignment cardinality constraint is conflicting between the relationship of users role assignment and the relationship of role permission assignment, trying to meet one of the two constraints may lead to violation of the others, constrained role mining problem is a NP complete problem [5], Harika proposes two different frameworks, one is to implement constraint after role mining, and the other is to implement constraint separately or simultaneously in the role mining process. One proposed a role mining method that are based permission cardinality constraint and user cardinality constraint, which merge roles by the similarity of roles to improve precision of roles' state, when it is running, it will consume substantial time and calculation about similarity of roles [11].

One proposed two heuristic role mining algorithms that satisfy both the permission cardinality constraint and the role usage cardinality constraints, since it randomly selects a row with the least number of users, the permission is defined as a role, so there is uncertainty, and the permission sets whose number is minimum is not necessarily a frequent item set [2]. The above method selects a constraint from the user role relationship and the role permission relationship to control the relationship between the user and the permission. The advantage of using role-based access control is that the role is used to bridge the gap between users and permission, users can get the required permission indirectly through the role. A user needs to remember all kinds of ID (identities) and passwords in multi-server environment [14], and outsourcing large-scale computing tasks to the cloud [10], so there are a large number of relationships between users and permission stored in enterprise. By implementing role-based access control, access control matrix can be simplified to facilitate enterprise storage and analysis.

The purpose of finding the smallest set of roles is to be able to express users and permissions with it. By mining association rules between permission. It can help define a role so that it can replace more relationship between users and permission. Take into account this consideration, the paper decided to design the algorithm based on the ideas of word frequent statistics and frequent item mining.

3 Constrained Role Mining Problem

Fundamental RBAC model has been introduced. We will only introduce some concepts about implementing the algorithm in this part.

Definition 1 (RBAC). *The Role Based Access Control (RBAC) model comprises the following components [17]:*

- $U = \{u_1, u_2, \ldots, u_n\}$, *$U$ represents user set;*
- $P = \{p_1, p_2, \ldots, p_n\}$, *$P$ represents permission set;*
- $R = \{r_1, r_2, \ldots, r_n\}$, *$R$ represents the role set;*
- $UA \subseteq U \times R$, *UA is a relationship from U to R;*
- $PA \subseteq R \times P$, *PA is a relationship from R to P.*

Definition 2 (Basic role mining problem, RMP). *Given a set of users U, a set of permission P and a user permission access control matrix UPA, find a group of roles R, user role assignment relationship UA and role permission assignment relationship PA. Satisfy $UA \otimes PA = UPA$, and minimizes $|R|$.*

Definition 3 (Permission cardinality constraint). *The permission cardinality constraint specifies the maximum number of permission a role can have in a RBAC system.*

Definition 4 (User cardinality constraint). *The user cardinality constraint is specified as the maximum number of user that a role can belong to in a RBAC system.*

Definition 5 (Role usage cardinality constraint). *The role usage cardinality constraint is defined as the limited number of roles which each user can have.*

Definition 6 (Permission assignment cardinality constraint). *The permission assignment cardinality constraint is defined as the limited number of roles that each permission can be assigned.*

When an enterprise establishes a RBAC system, the relationship of user and permission will change with the development of the enterprise. Therefore, the company

should control user cardinality constraint and role usage cardinality constraint in a dynamic process. The permission assignment cardinality constraint does not need to be considered, because the purpose of controlling the permission assignment cardinality can be achieved by controlling the role usage cardinality constraint. The permission cardinality constraint could satisfy the separation of duties constraints to prevent roles from having mutually exclusive permission. When a user accesses the allocation of resources in the cloud, if the number of permission of the role is limited, the hierarchical management of the role can be facilitated [20]. So the algorithms proposed in this paper is to mine the role in the access control matrix meeting the permission cardinality constraint.

Definition 7 (Weighted Structural Complexity, WSC)**.** *[13]. Given* $W =< w_r, w_u, w_p, w_h, w_d >$*, where* $w_r, w_u, w_p, w_h, w_d \in Q^+ \cup \{\infty\}$*, the Weighted Structure Complexity (WSC) and RBAC state* γ*, which is denoted as* $wsc(\gamma, w)$*, is computed as follows. We have*

$$\begin{aligned} wsc(\gamma, w) \quad = \quad & w_r * |R| + w_u * |UA| + \\ & w_p * |PA| + w_h * |t\_reduce(RH)| + \\ & w_d * |DUPA| \end{aligned} \tag{1}$$

Where $|\cdot|$ denotes the size of the set or relations, and $t_reduce(RH)$denotes the transitive reduction of the role-hierarchy.Since the proposed algorithm does not consider the role in heritance, and does not allow the authority to be directly assigned to the user, by setting $w_r = w_u = w_p = 1, w_h = w_d = \infty$. Arithmetic involving ∞ is defined as follows: $0 * \infty = 0, \forall x \in Q^+ \; x * \infty = \infty, \forall x \in Q \cup \{\infty\} \; x + \infty = \infty$.

Definition 8 (Role Mining Problem Satisfied the Permission Cardinality Constraint)**.** *Given a set of user* $U = \{u_1, u_2, \cdots, u_n\}$*, a set of permission* $P = \{p_1, p_2, \cdots, p_n\}$*, a user permission access control matrix UPA, and a positive integer* $t, t > 1$*. Find a group of roles* $R = \{r_1, r_2, \cdots, r_q\}$*, a user role assignment relationship UA and a role permission assignment relationship PA, Satisfy* $UA \otimes PA = UPA$*,* $\forall r_i \in R \; |PermsR(r_i)| \leq t$*,* $1 \leq i \leq q$*,* $PermsR(r_i)$ *represents the permission that role* r_i *own, and minimize WSC.*

The evaluation goal of role mining cannot be measured by reducing the number of the roles.Because simply reducing the number of roles, in order to achieve, finally lead to the increment of UA and PA. Taking the reduction of WSC as the final metric can fully measure the definitive role mining situation.

4 Algorithm Overview

In machine learning algorithms, FP-growth algorithm is an algorithm founded on association rules.It can mine a group of items with a strong correlation.So it could be used to basket analysis,merchant arranges placement of goods conveniently and bundled sale of goods.Mapping the user permission access control matrix to the user's shopping list, where the permission represents the product. If you use FP-growth algorithm, you can mine the frequent permission set in the access control matrix and define the frequent item set as a role. Where the permission cardinality constraint can be defined as the permission set has at most several permission, so as to achieve the purpose of satisfying the defined cardinality constraint.When the data set is large, the FP-growth algorithm recursively generates a large number of conditional pattern libraries and conditional FP-tree. In this situation, the algorithm needs excessive memory and has low efficiency [4]. Additionally,a role mining algorithm as a method for frequent item mining. It becomes less efficient due to the larger set of roles and multiple iterations of the FP-growth algorithm.This paper presents a method of clustering frequent permission sets in the access control matrix, clustering similar permission sets, and generating role sets by iterative simplification. Finally, the set of roles corresponding to each user is found in the access control matrix by using the generated set of roles.

4.1 Role Mining Algorithm Satisfied the Permission Cardinality Constraint

Algorithm 1 Role mining algorithm based on word frequency statistics

Input: Access control matrix:UPA,Permission Cardinality:$Limited$.

Output: Role set:R, the relationship of user and role:UA,WSC.

1: Begin
2: Column of UPA is sorted from left to right in descending order according to the number of users of the permission.
3: Sort the rows of UPA from top to bottom in descending order,depending on the location of the permission of UPA.If the user has permission to the front left,it will be ranked first.
4: The location which is 1 is replaced by the original order of the permission in UPA.
5: Define every $Limited$ permission from left to right for each user in UPA as a role.
6: Define each role as a key in a key-value pair whose number of occurrences is defined as the value of the key.
7: Generate user role relationships (UA) based on generated role sets (R) and access control relationship F(UPA).
8: $WSC = |UA| + |PA| + |R|$, UA represents the role set.
9: End

By sorting the rows and columns of UPA, it is possible to have similar permission gathered together.Defining every *limited* permissions for each user in the access control as a role, which not only meets permission cardinal-

ity constraint, but also limits the number of each user's role. In addition, the idea of the algorithm draws on word frequency statistics.By sorting the UPA,each row of the UPA can be treated as a string of characters,and the permission cardinality constraint is the number of characters that specify the intercepted zero-free word (the user does not get the permission).

Algorithm 2 Iterative based role mining algorithm

Input: Access control matrix:UPA,Permission Cardinality:$Limited$,Iterative benchmark:$IteraBench$.

Output: Role set:R, the relationship of user and role:UA,WSC.

1: Begin
2: Column of UPA is sorted from left to right in descending order according to the number of users of the permission.
3: Sort the rows of UPA from top to bottom in descending order,depending on the location of the permission of UPA.The user has permission to the front left,which will be ranked first.
4: The location which is 1 is replaced by the original order of permission in UPA.
5: Define the maximum intersection of the permission of neighboring users in the matrix as a role.
6: Sort the generated role set according to the number of permission included, and deletes the smaller role.
7: Select a role greater than the iteration cardinality line ($IteraBench$) from the bottom to the top of the matrix as a temporary role $r^{'}$.
8: $\forall R \supset r^{'}, R = R - r^{'}$
9: Sort *candidate role set*.
10: Replace the relationship between users and roles in the access control matrix with candidate role set.
11: **if** There is a user's remaining permission which cannot be replaced by the roles in *candidate role set* **then**
12: Generate a role to add to *candidate role set* for each user's remaining permissions that cannot be replaced with roles in role set,
13: **goto** 7
14: **else**
15: Split the role who's the number of permission greater than $Limited$ and generate some new roles.
16: **end if**
17: Generate user role relationships (UA) based on generated role sets (R) and access control relationship (UPA)
18: $WSC = |UA| + |PA| + |R|$, UA is the role set.
19: End

Algorithm 2 also is required to perform the same sorting as algorithm 1. The intersection of the permissions owned by each user and neighboring users' is to extract the largest identical portion locally. The global optimum is accomplished as much as possible by local optimization. Iterative benchmark defines the criteria for iteration in the process of role set reduction,if it is too large,it is easy to violate the permission cardinality constraint. In the 6th step, if the size of the candidate role is less than 4 and its frequency is less than 2, we will delete it.About the permission owned by a small number of users, it is likely to be a key permission, so this article also defines it as a role, when assigning, it should limit the assigned amount of such roles and prevent unauthorized operation.In order to ease the understanding of the processing of the two algorithms, we present flow charts of the two algorithms. As showed in Figure 1 and Figure 2, We can see from the figure that the two algorithms are the same ones at the beginning and end, but the access matrix processing method after sorting is different. It is worth mentioning that in fact, the content of the loop in the flowchart of algorithm 2 will only be executed once. Because the process is performed only once, the remaining permissions of all users are generated by the relevant roles, and the loop will not be run.

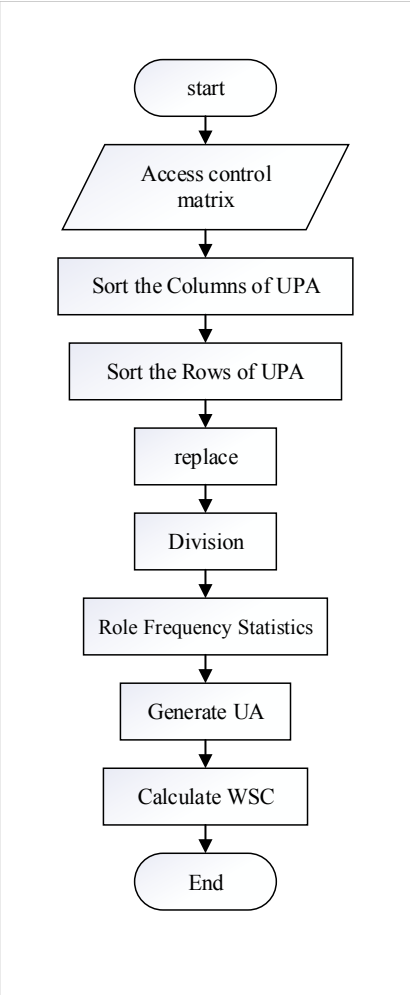


Figure 1: The flow chart of Algorithm 1

4.2 Analysis of Algorithms

We will illustrate the general operation of the algorithms, for example, Table 1 is an access control matrix. After column sorting and row sorting, it will produce the matrix

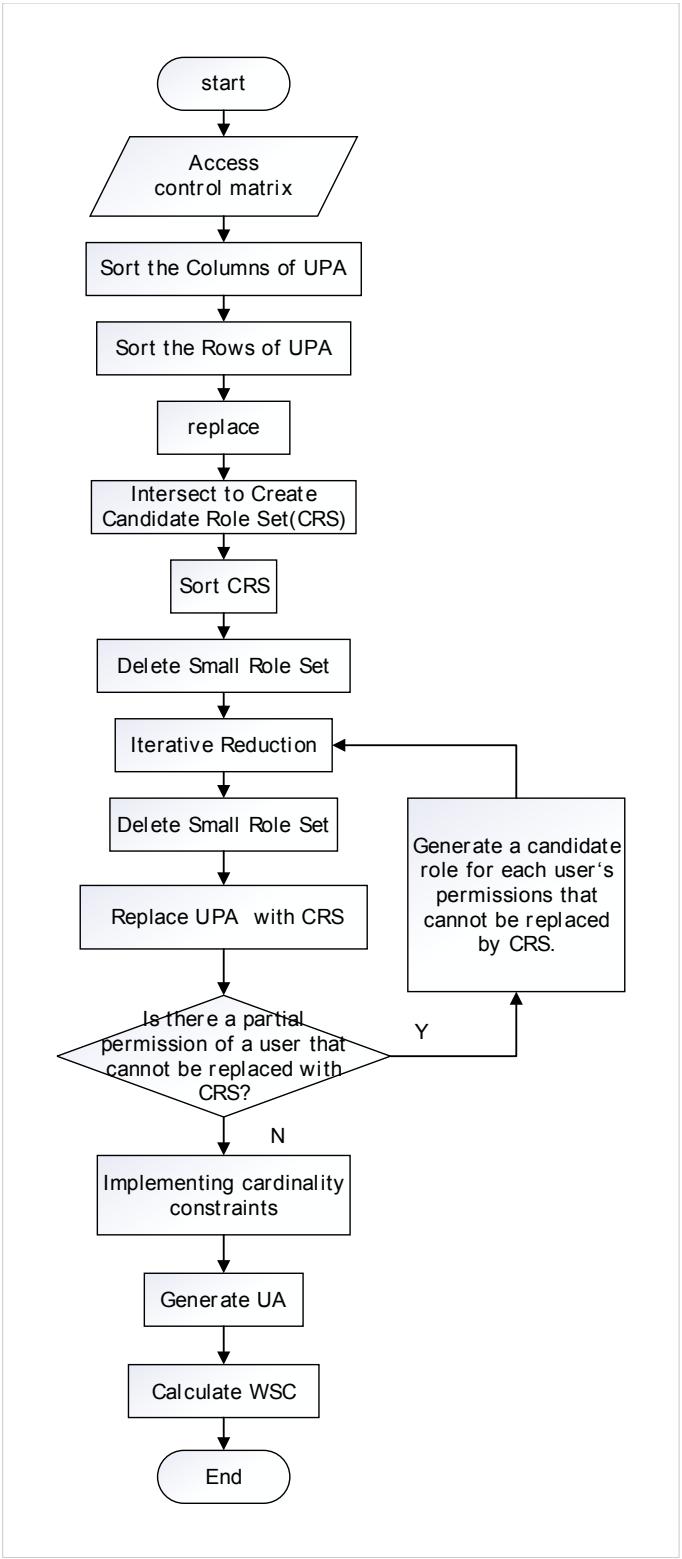


Figure 2: The flow chart of Algorithm 2

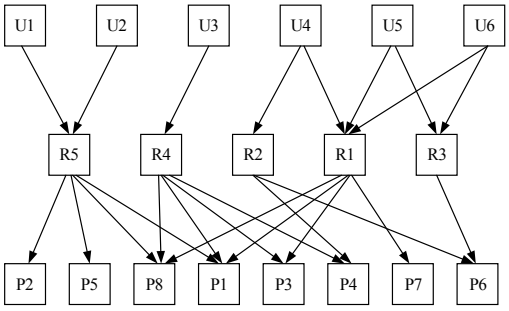


Figure 3: The result of Algorithm 1

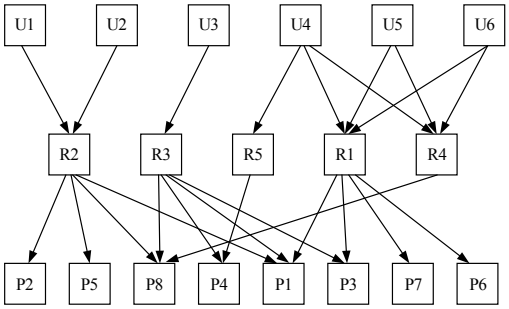


Figure 4: The result of Algorithm 2

shown in Table 2, The number in the matrix is the number of the column's permission, From the Table, we can see that sorting makes the original disordered access control matrix more regular. It is more convenient to mine roles on the matrix. Algorithm 2 mine the roles from the perspective of the line of the UPA. In this example,the permission cardianlity is set to four, The relationship between users,roles and permissions through algorithm 1 is shown in Figure 3, Each segmented word represents a role. Because a role in a candidate role set might be a subset of another role, we set a benchmark for iterative reduction. The number of inclusion relationships between roles decreases by reduction. The relationship between users, roles and permission through algorithm 2 is given in Figure 4. The time complexity is required for algorithm 1 to sort the access control matrix once is $O(c\log_2 c)$,where c represents the number of columns in the access control matrix. The time complexity required to perform a row sort is $O(r\log_2 r)$, where r represents the number of rows in the access control matrix. Furthermore, generating a set of roles requires traversing the access control matrix once, and generating a user role assignment relationship also needs to be traversed once. Therefore the total time complexity of algorithm 1 is $O(c\log_2 c+r\log_2 r+2cr)$.The spatial complexity of algorithm 1 is $O(cr+|UA|+|PA|)$. UA represents user role relationship. PA represents role permission relationship.

Algorithm 2 also needs to first sort the row and column of the access control matrix, and the time complexity is $O(c\log_2 c+r\log_2 r)$, Each row in the access control needs to produce an intersection with the adjacent row, which needs to be compared $2c(r-2)$ times. Multiple iterative

Table 1: Access control matrix

user	P1	P2	P3	P4	P5	P6	P7	P8
U1	1	1	0	0	1	0	0	1
U2	1	1	0	0	1	0	0	1
U3	1	0	1	1	0	0	0	1
U4	1	0	1	1	0	1	1	1
U5	1	0	1	0	0	1	1	1
U6	1	0	1	0	0	1	1	1

Table 2: Sorted access control matrix

user	P1	P8	P3	P7	P6	P5	P4	P2
U4	1	8	3	7	6	0	4	0
U5	1	8	3	7	6	0	0	0
U6	1	8	3	7	6	0	0	0
U3	1	8	3	0	0	0	4	0
U1	1	8	0	0	0	5	0	2
U2	1	8	0	0	0	5	0	2

reduction occurs when the final role set is generated. In the process of iteration, only if the role's number of permission is greater than the given value (that is greater than the IteraBench defined in the algorithm 2), the role can be iterated and simplified with the previous roles in the array of role sets. So the time complexity of the part is approximately $O(|R|cr)$. At last, generating user role relationships needs to be compared $|PA|cr$ time. Consequently, the total time complexity of algorithm 2 is approximately $O(n\log_2 n + (1 + |R| + |PA|)cr)$, n is the maximum of c and r. The spatial complexity of algorithm 2 is $O(|UA| + |PA| + cr)$. There are a few places in algorithm 1 and algorithm 2 which can be changed to parallel operations, such as the division of permissions for each user and the generation of intersections of adjacent rows.

5 Experimental Evaluation

In the following sections, we present the experimental evaluation of our algorithms and PRUCC-RM which is proposed by Blundo and satisfies the permission cardinality constraint [2]. The test platform hardware is 3.4Ghz Intel CPU and 8 GB memories. The operation system is Windows 7, the program is run in a VMware virtual machine, the operating system image used is Ubuntu, and its version number is 16.04 LTS. In the Spark pseudo-distributed cluster, using Scala high-level programming language to perform programming experiments in Intellij IDEA.

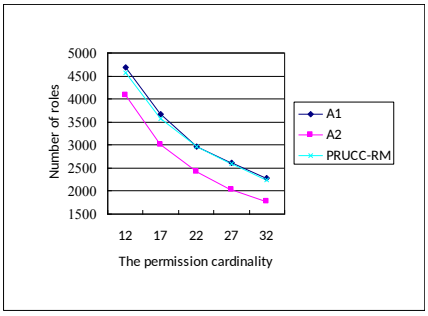
In order to be possible to repeat this experiment and the data sets used are all public data sets, which have been used in the literature [2, 5, 8]. The URL for downloading the role mining tool RMiner is given there [9]. Through this website, not only you download the public data set, but you can also download the role mining tool RMiner. Table 3 lists sizes, execution time and defined iteration benchmark for each data set, where userMaxPerm represents the maximum number of permissions a user has in the data set. *IteraBench* is the threshold for each iteration of different data sets, and the reader can choose the threshold of the iteration. Finally, we also represent the execution time of three algorithms in the worst case.Algorithm 2 takes longer than algorithm 1 because it requires multiple iterations. Since PRUCC-RM does not have row and column ordering, it takes the least amount of time. When the data set is small, the running time of the three algorithms is approximately equal. Defining algorithm 1 as A1 and algorithm 2 as A2.

From Figure 5 to Figure 12, it can be observed that as the permission cardianlity increases, the number of roles generated by the three algorithms is gradually decreasing. Because the maximum number of permission each role can have, resulting in some users have all the permission which can be replaced by fewer roles.Some data sets appear with the increase of the permission cardinality, and the number of roles subsequently grows. Algorithm 1 divides the permission of each user according to the permission cardinality. When the permission cardianlity is set to an appropriate value, the number of generated roles is minimized. When the permission cardianlity exceeds the appropriate value, more roles are generated to satisfy the relationship between the user and the permission. In terms of weighted structural complexity, the overall trend of algorithm 1 decrease first and then increases with the increase of the permission cardinality. Because as the number of the permission cardinality increases, the number of roles decreases overall, while the size of $|PA|$ may increase, and the size of $|UA|$ may decrease. When a suit-

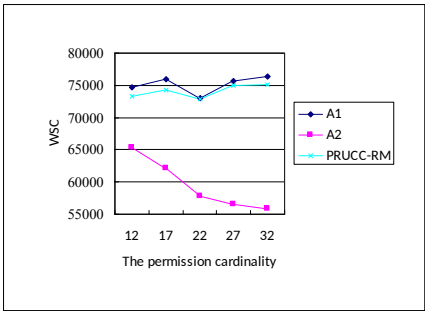
Table 3: Characteristics of real data sets

Data Set	|U|	|P|	userMaxPerm	IteraBench	Execution time(s)		
					A1	A2	PRUCC-RM
Americas_Large	3485	10127	733	10	251.541	273.533	24.113
Americas_Small	3477	1587	310	13	20.179	24.968	11.006
Apj	2044	1164	58	12	11.531	12.258	9.701
Emea	35	3046	554	10	7.444	7.778	7.231
Healthcare	46	46	46	10	7.176	7.132	7.165
Domino	79	231	209	10	7.058	7.310	7.385
Firewall1	365	709	617	15	7.841	8.137	7.599
Firewall2	325	590	590	12	7.443	7.827	7.875

[1] `http://code.google.com/p/rminer/`

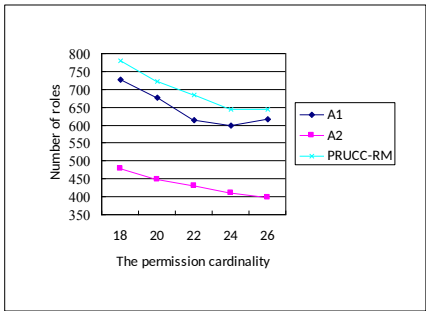


(a) Number of The Roles

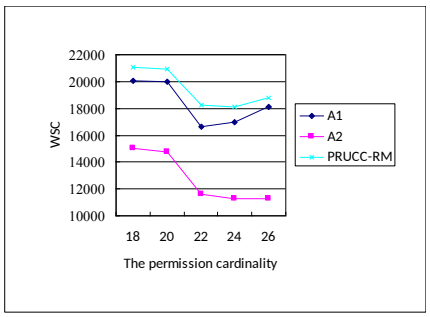


(b) WSC

Figure 5: Experimental results of three algorithms in Americas_Large

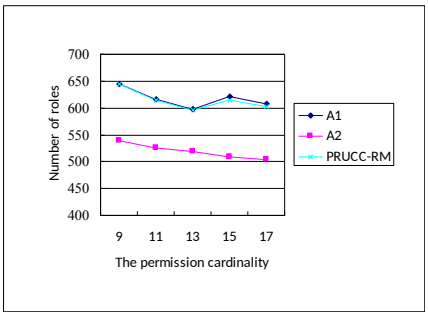


(a) Number of The Roles

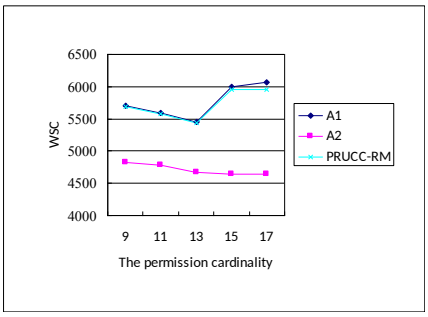


(b) WSC

Figure 6: Experimental results of three algorithms in Americas_Small

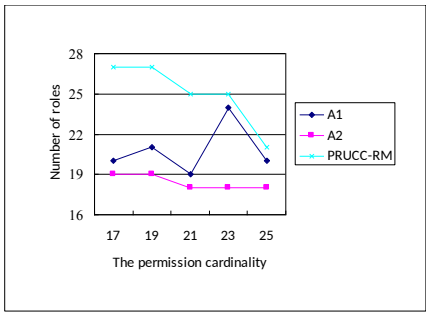


(a) Number of The Roles

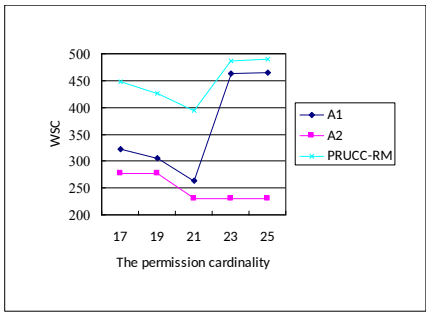


(b) WSC

Figure 7: Experimental results of three algorithms in Apj

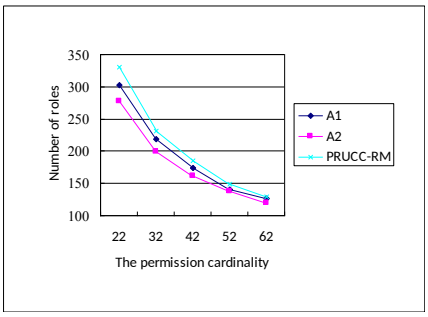


(a) Number of The Roles

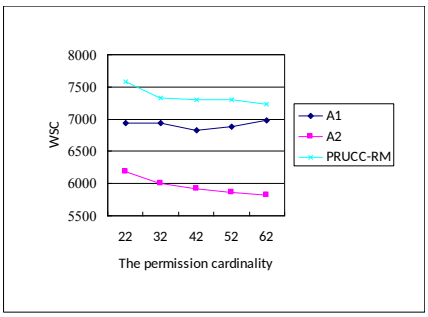


(b) WSC

Figure 9: Experimental results of three algorithms in Healthcare

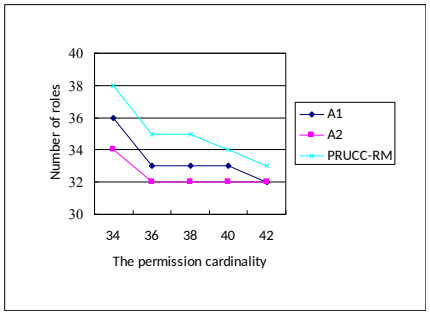


(a) Number of The Roles

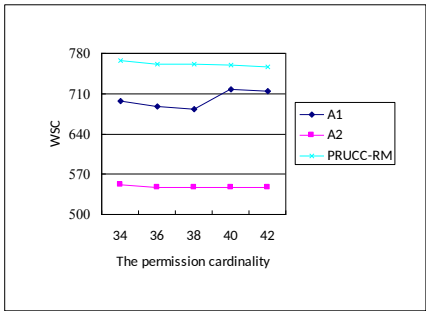


(b) WSC

Figure 8: Experimental results of three algorithms in Emea

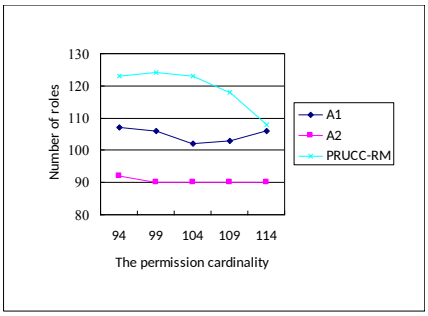


(a) Number of The Roles

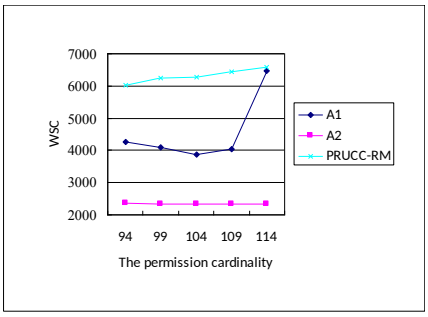


(b) WSC

Figure 10: Experimental results of three algorithms in Domino

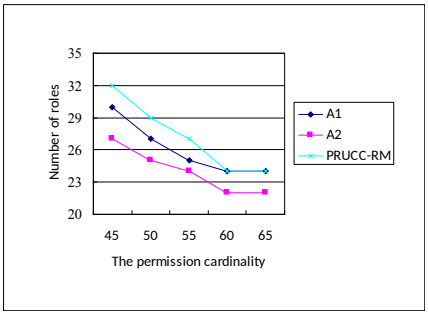


(a) Number of The Roles

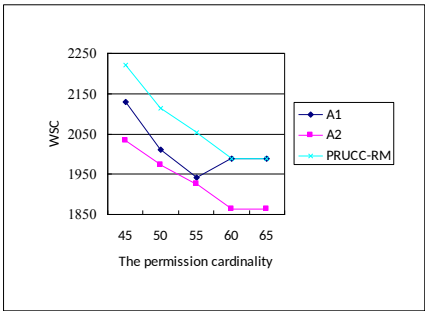


(b) WSC

Figure 11: Experimental results of three algorithms in Firewall1



(a) Number of The Roles



(b) WSC

Figure 12: Experimental results of three algorithms in Firewall2

able permission cardinality is reached, the weighted structural complexity reaches a minimum.The performance of PRUCC-RM is similar to algorithm 1, the biggest difference is that PRUCC-RM has no sequence of rows and columns. The overall tendency of algorithm 2 decreases with the increase of the permission cardinality. Because the permission cardinality is increased, Algorithm 2 can be better mine more frequent permission sets to define as roles, so the overall weighted structural complexity is reduced. Figure 7 and Figure 5 show that the performance of PRUCC-RM is between algorithm 1 and algorithm 2. Because the results produced by PRUCC-RM are related to the location of the permission in UPA, different results are produced when adjusting the position of each permission (*i.e.*, each column) is adjusted in the UPA. Algorithm 1 needs to be sorted by row and column, so it has nothing to do with the position of the permission in UPA.

6 Conclusion

In this paper, we have proposed two algorithms to solve the role mining problem meeting the permission cardianlity constraint in the public data set.Compared with PRUCC-RM. The experimental results demonstrate that the second algorithm performs better than algorithm 1 and PRUCC-RM. However, the algorithm 1 is more simple. The two algorithms proposed in this paper can be applied in the field of frequent item mining. In addition, we can also define the candidate roles with higher frequency as the definitive role, and the roles owned by a small number of users can be managed separately because they are either very important roles or unimportant. Any kind of role mining method has its limitations. Algorithm 2 proposed in this paper is an open role mining method, and the iteration benchmark in the running process of the program can be debugged by the user. The permission cardinality constraint can be adjusted according to the requirements of the number of permissions that the role in the enterprise.Practical experience shows that RBAC is very suitable for systems where the relationship between users and permission does not vary frequently.Users' circumstance and the business process should also be considered after role mining.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (No.61662056) and Inner Mongolia Natural Science Foundation of China (No.2016MS0608, No.2016MS0609). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. V. Belim and A. N. Mironenko, "Using the graph-theoretic approach to solving the role mining prob-

lem," in *2018 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, pp. 1–5, Omsk, Russia, Nov 2018.

[2] C. Blundo, S. Cimato, and L. Siniscalchi, "Prucc-RM: Permission-role-usage cardinality constrained role mining," in *IEEE Computer Software and Applications Conference*, pp. 149–154, Torino, Italy, July 2017.

[3] L. Dong, Y. Wang, R. Liu, B. Pi, and L. Wu, "Toward edge minability for role mining in bipartite networks," *Physica A: Statistical Mechanics and its Applications*, vol. 462, pp. 274–286, 2016.

[4] C. Fu, X. Wang, L. Zhang, and L. Qiao, "Mining algorithm for association rules in big data based on hadoop," *AIP Conference Proceedings*, vol. 1955, no. 1, p. 040035, 2018.

[5] P. Harika, M. Nagajyothi, J. C. John, S. Sural, J. Vaidya, and V. Atluri, "Meeting cardinality constraints in role mining," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 71–84, 2015.

[6] H. Huang, S. Feng, J. Liu, and H. Du, "Handling least privilege problem and role mining in rbac," *Journal of Combinatorial Optimization*, vol. 30, no. 1, pp. 63–86, 2015.

[7] J. Jiang, X. Yuan, and R. Mao, "Research on role mining algorithms in rbac," in *Proceedings of the 2018 2nd High Performance Computing and Cluster Technologies Conference*, HPCCT 2018, pp. 1–5, Beijing, China, June 2018.

[8] R. Kumar, S. Sural, and A. Gupta, "Mining rbac roles under cardinality constraint," in *International Conference on Information Systems Security*, pp. 171–185, Gandhinagar, India, Dec 2010.

[9] R. Li, H. Li, W. Wei, X. Ma, and X. Gu, "Rminer: A tool set for role mining," in *Acm Symposium on Access Control Models and Technologies*, pp. 193–196, Amsterdam, The Netherlands, June 2013.

[10] L. Liu, Z. Cao, and C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, pp. 29–35, Sep 2018.

[11] X. Ma, R. Li, H. Wang, and H. Li, "Role mining based on permission cardinality constraint and user cardinality constraint," *Security and Communication Networks*, vol. 8, no. 13, pp. 2317–2328, 2015.

[12] B. Mitra, S. Sural, J. Vaidya, and V. Atluri, "A survey of role mining," *Acm Computing Surveys*, vol. 48, no. 4, pp. 1–37, 2016.

[13] I. Molloy, C. Hong, T. Li, Q. Wang, N. Li, E. Bertino, S. B. Calo, and J. Lobo, "Mining roles with multiple objectives," *Acm Transactions on Information and System Security*, vol. 13, no. 4, pp. 1–35, 2010.

[14] H. T. Pan, C. S. Pan, S. C. Tsaur, and M. S. Hwang, "Cryptanalysis of efficient dynamic id based remote user authentication scheme in multi-server environment using smart card," in *Proceedings - 12th International Conference on Computational Intelligence and Security, CIS 2016*, pp. 590–593, Wuxi, Jiangsu, China, Dec 2016.

[15] U. P. Rao and N. K. Singh, "Weighted role based data dependency approach for intrusion detection in database," *International Journal of Network Security*, vol. 19, no. 3, pp. 358–370, 2017.

[16] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[17] P. Sarana, A. Roy, S. Sural, J. Vaidya, and V. Atluri, "Role mining in the presence of separation of duty constraints," in *Information Systems Security* (Sushil Jajoda and Chandan Mazumdar, eds.), pp. 98–117, Cham, 2015. Springer International Publishing.

[18] Y. Tian, Y. Peng, G. Gao, and X. Peng, "Role-based access control for body area networks using attribute-based encryption in cloud storage," *International Journal of Network Security*, vol. 19, no. 5, pp. 720–726, 2017.

[19] J. Vaidya, V. Atluri, and G. Qi, "The role mining problem: A formal perspective.," *Acm Transactions on Information and System Security*, vol. 13, no. 3, pp. 1–31, 2010.

[20] J. Wang, J. Liu, and H. Zhang, "Access control based resource allocation in cloud computing environment," *International Journal of Network Security*, vol. 19, no. 2, pp. 236–243, 2017.

[21] W. Ye, R. Li, X. Gu, Y. Li, and K. Wen, "Role mining using answer set programming," *Future Generation Computer Systems*, vol. 55, pp. 336–343, 2016.

Biography

Jingyu Wang is a Ph.D and professor in school of Information Engineering, Inner Mongolia University of Science and Technology, member of the Chinese Computer Society, and a member of the ACM Institute. His main research interests include Big data access control, information security and data mining. Contact him at 13734728816@126.com.

Jingnan Dong is a postgraduate in school of Information Engineering, Inner Mongolia University of Science and Technology,His main research interests include big data, data mining and cloud computing. Contact him at 867324154@qq.com.

Yuesheng Tan is a postgraduate and professor at the School of Information Engineering, Inner Mongolia University of Science and Technology, member of the Chinese Computer Society. His main research interests include high performance computing cloud computing, large-scale data processing and mining. Contact him at 13604729678@139.com.

Detecting Improper Behaviors of Stubbornly Requesting Permissions in Android Applications

Jianmeng Huang, Wenchao Huang, Fuyou Miao, and Yan Xiong
(Corresponding author: Wenchao Huang)

School of Computer Science and Technology, University of Science and Technology of China
Elec-3 (Diansan) Building, West Campus of USTC, Huang Shan Road, Hefei, Anhui Province, China
(Email: huangwc@ustc.edu.cn)

(Received June 13, 2018; Revised and Accepted Nov. 22, 2018; First Online July 16, 2019)

Abstract

Android applications may stubbornly request permissions at initialization: if the user does not grant the requested permissions, these applications would simply exit, refusing to provide any functionalities. As a result, users are urged by this behavior to grant sensitive permissions and users actually lose the power to control their sensitive data, which may cause permission abuse and privacy leakage. In this paper, we propose an approach to automatically detect the improper behaviors of stubbornly requesting permissions. Experiments on real-world applications demonstrate the effectiveness of our approach and reveal that almost 24% analyzed applications contain stubborn permission requests.

Keywords: Android Security; Improper Behaviors Detection; Privacy Leakage; Stubborn Permission Requests

1 Introduction

The dramatic growth of Android applications (*apps* for short) has raised significant security concerns. Android dominated the smart phone market with a share of 85% in 2017 [14]. Most of the apps provide functionalities relying on sensitive user data (such as SMS and contacts), as well as certain system features (such as camera and microphone). However, a number of malicious apps abuse their privileges on private data [9], which threatens users' privacy.

To protect users' privacy, the Android permission mechanism [10] provides control on whether an app is allowed to access certain sensitive resources. On Android 6.0 and higher versions, if an app wants to access sensitive data, it should request corresponding permissions at runtime [7]. The permission mechanism does restrict the improper behaviors to a certain extent. For example, if a user does not need the location services from an app or does not trust the app, the user can deny the permission of accessing GPS to this app. Overall, the newer Android permission mechanism helps users to protect certain sensitive data by granting or denying corresponding permissions at runtime, allowing that only the functionalities related to the denied permissions are restricted.

However, some apps may get their requested permissions by urging users to grant them: if users do not grant the permissions, these apps would exit. Two reasons may account for the purposes of this behavior. First, these apps intentionally collect user's sensitive data. They gain profit from user's privacy, so they need to get corresponding permissions. Second, to reduce workload, the developers of these apps have not implement the codes which handle the exception of not having sensitive permissions. Hence, these apps are stubborn to get the permissions, otherwise the apps may crash. In this paper, we call this behavior as the stubborn permission request, which actually puts pressure on users to grant the requested permissions.

Unfortunately, users are vulnerable to the stubborn permission requests. On the one hand, users are not aware of the stubborn permission request in the app before the app is installed and running. After the app is installed, users may bother to uninstall it or choose another app with similar functionalities. On the other hand, users may be attracted by some functionalities of the app, which makes users ignore the risk of privacy leakage and hence grant the permissions. A survey [11] of 308 Android users and a laboratory study of 25 Android users found that only 17 percent paid attention to permissions. If users yield to the stubborn requests, their privacy is under threat for that the app would get full access to its required sensitive data. At runtime, users are not aware of when and which permissions are used. Therefore, it is necessary to inform users about the stubborn permission requests of an app before the app is installed into the device.

In this paper, we present an approach based on static analysis to detect improper behaviors of stubbornly requesting permissions in Android apps. We first study and model the behaviors of stubborn permission requests: if the requested permissions are not granted, the app would

exit. Then we statically identify the behaviors in the decompiled codes. To the best of our knowledge, our approach is the first to detect stubborn permission requests in Android apps. The experimental result shows that 24% of the tested popular apps contain stubborn permission requests. Such result indicates that users who are using these apps are exposed to the potential risk of privacy leakage, and the app market has not noticed the risk of such improper behavior.

Our approach can be adopted by Android app markets. The app market could add a label to the app which contains stubborn permission requests. As a result, users could be warned by the label. If a user does not trust the app, he/she could choose to install other alternative apps. With the result of our detection, the market could also conduct further detailed analysis on the apps in order to figure out how the sensitive data are used.

The main contributions of this paper are as follows:

- We study the stubborn permission requests in Android apps. To the best of our knowledge, our work is the first to investigate this kind of behaviors.
- We propose a static analysis approach to detect stubborn permission requests in Android apps.
- We demonstrate the effectiveness of our approach and present our findings.

The rest of this paper is organized as follows: Section 2 describes the background and the current problem. Section 3 presents the design of our approach and Section 4 describes experimental results. Section 5 describes related work and Section 6 concludes the paper.

2 Background and Problem Statement

2.1 Android Permissions

The main purpose of Android permission mechanism is to protect Android end users' privacy. In the Android system, every app runs in a limited-access sandbox. If an app needs to use resources or information outside of its sandbox, the app has to request the appropriate permissions (*e.g.*, contacts, Bluetooth and location). Before Android 6.0, the permissions are declared in the app manifest file and granted by users at install time. This permission mechanism does protect users' privacy to a certain extent. However, the permission model cannot be easily deployed, for that before appropriately using the Android permission model against suspicious apps, users should understand the program behaviors of the apps.

On Android 6.0 and higher versions, the system grants the permission automatically or might prompt the user to approve the request, depending on the level of the permission. Particularly, the Android permissions are categorized into two levels: the normal and dangerous permissions [8]. Some permissions are considered "normal" (*e.g.*, Internet, setting alarm and NFC) so the system immediately grants them upon installation. Other permissions are considered "dangerous" (*e.g.*, SMS, camera and storage) so that apps must explicitly request for users' agreements at runtime. Only dangerous permissions require users' agreements. The proper way of utilizing the dangerous permissions is:

1) Adding the permissions to the manifest;
2) If an app needs a dangerous permission, it must check whether it has that permission every time it performs an operation that requires this permission, because the user can revoke the permission from the app at any time;
3) If the app does not have the permission, the app must prompt the user for that permission using certain APIs provided by Android, which brings up a standard Android dialog that cannot be customized by developers.

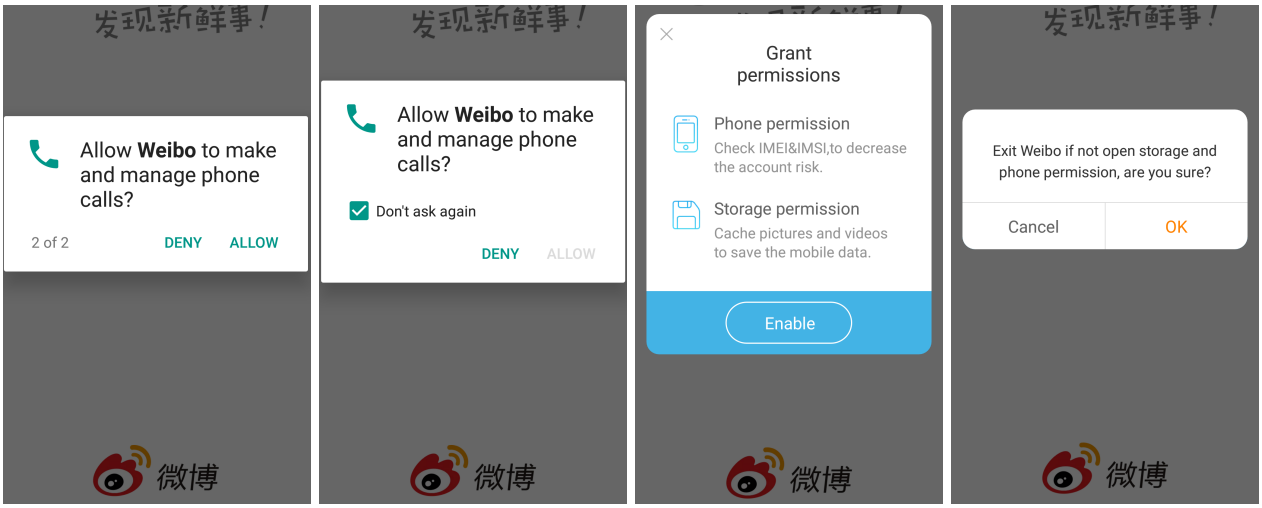
Besides, the permission request should occur at the time that the operation needs the corresponding sensitive resource, so that the user could understand why the app needs the permission in that circumstance and grants the permission to the app if the request is reasonable. After the user responds to an app's permission request, the system invokes method `onRequestPermissionsResult()` in the app, passing the user's response to the app. The app should override this method to find out whether the permission is granted.

The new permission mechanism further organizes permissions into groups related to a device's capabilities or features. Under this mechanism, permission requests are handled at the group level and a single permission group corresponds to several permission declarations in the app manifest. For example, the SMS group includes both the `READ_SMS` and the `RECEIVE_SMS` declarations. If an app gets the permission of `SMS` group, it has both the two permissions in this group.

2.2 Problem Statement

The permission mechanism on Android 6.0 and higher versions helps users better understand the behaviors of using sensitive permissions in an app and hence is more effective in protecting users' privacy. However, proper deployment of the permission mechanism requires higher workload of developers, since the app should always check for and request permissions at runtime to guard against runtime errors. Even if the user grants an app dangerous permissions, the app cannot always rely on having them. Because the user has the option to disable permissions in system settings.

Some apps urge users to grant the requested permissions when these apps begin to run. If the app has the dangerous permissions, the app would continue to run; Otherwise, the app would prompt the user for the permissions. If the user refuses to grant the permissions,



(a) The first permission request. (b) The second permission request. (c) The explanation of permission usage. (d) The final notification.

Figure 1: The permission request of an stubborn app.

the app would immediately exit. Figure 1 illustrates a real-world app which would exit if it does not get the required permissions. Figure 1a and Figure 1b show the standard Android dialogs which are used for requesting permissions. If the user denies the request twice, and chooses "Don't ask again", the dialog would not show at following runs of the app. However, this app prompts its own dialog (Figure 1c) which tells the users how to grant the required permissions in system settings. If the user denies all these requests, this app would inform the user that it would exit if not granted the permissions (Figure 1d).

The permission requests at the initialization of an app divorce from the original intention of the new Android permission mechanism and stubborn permission requests (*i.e.*, if the user does not grant the permissions, the app would exit) hurt the user's rights of utilizing functionalities which do not require dangerous permissions. It is reasonable for the app to disable the functionality that relies on the required permission (not granted), but it is not reasonable for the app to disable all the functionalities if it does not get all of the requested permission.

Challenge: The challenge of automatically detecting stubborn permission requests in Android apps is how to precisely and concisely model this kind of improper behaviors. To the best of our knowledge, the behaviors of stubbornly requesting permissions have not been investigated by existing researches.

Hence, adequate investigations from Android apps are required. Then, to automatically detect such improper behaviors in Android apps, a precise and concise model should be proposed. First, the model should be precise so that the detection result could achieve high precision. Second, the model should be concise so that the detection could be efficient.

3 Design and Implement

To detect the stubborn permission requests in Android applications, we propose a static analysis approach to find situations that if not granted required permissions, the app exits. This section describes our design and implement.

3.1 Overview

Our insight of detecting the improper of stubbornly requesting permission is based on the observation that all such kind of improper behaviors share the same subprocess that if the requesting result is "not granted", the app would finally finish all the activities. Hence, our detection is implemented by searching such patterns in the app.

Definition 1. *A **stubborn permission request** is an improper app behavior that if the user does not grant an app requested dangerous permissions, the app would not provide any functionality, including functionalities which do not need the requested permissions. Besides, when the app is started next time, the app would check whether it has certain dangerous permissions. If not, the app would request for permissions again.*

Definition 1 describes the improper behavior of stubbornly requesting permissions researched by this paper. We argue that the permission requesting prompt should provide users a choice of whether to share privacy with the app, rather than urging users to grant the permissions.

Based on the above definition, our detection for stubborn permission requests includes two steps. As the stubborn permission requests begin with checking permissions, our first step locates code fragments of checking whether an app has dangerous permissions, which is used as the entry points of our further analyzing. The second step figures out whether the code fragments of entry

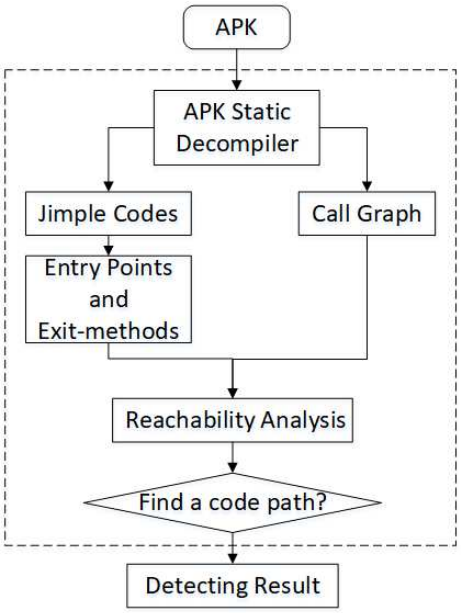


Figure 2: Framework of our approach

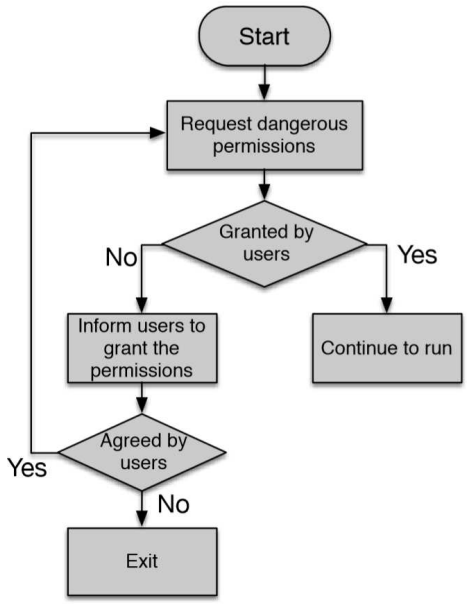


Figure 3: Flow of a stubborn permission request

points would finally lead to the exit of the app. If there exist code paths between the code fragments of checking dangerous permissions and finishing activities, which means that in certain conditions (*e.g.*, the user denies the permission requests), the app would exit after checking whether it has the dangerous permissions, and the stubborn permission requesting behavior is found.

3.2 Design

In this subsection, we first introduce how we choose the entry points for our further analyzing. Then we describe how we figure out whether the entry points would lead to the exit of an app. Figure 2 shows the methodology of our approach. Particularly, we use Soot [18] as the underlying analysis infrastructure. Soot translates the bytecode of an app to Jimple representation, a statement based intermediate representation, and it can generate an accurate call graph of the app. A call graph is a control flow graph which represents calling relationships between methods in an app. Then our approach locates the entry points and exit-methods (APIs which cause apps to exit). Finally, we apply a reachability analysis to figure out whether there exists a code path between the entry points and exit-methods. If found, we report the improper behavior of stubborn permission requests. Similar to prior static approaches [1,20], our analysis is based on the Jimple representation and the call graph. We do not adopt dynamic analysis approach because dynamic analysis faces the problem of low testing coverage, which causes insufficient analysis of the app.

3.2.1 Entry Points and Exit-methods

As our approach focuses on detecting stubborn permission requests, we first figure out the processes of stubborn permission requests. Figure 3 shows the normal flow of a stubborn permission request. The app asks for dangerous permissions first, then the user makes a choice of denying it or allowing it. Afterwards, the app could get the requesting result from method `onRequestPermissionsResult()`. If the requested permission is not granted, the app would inform the user to grant the permission by a prompted dialog or through the system setting. Finally, if the user still refuses to grant the requested dangerous permissions, the app would exit. Overall, whether the app is going to exit depends on the results of method `onRequestPermissionsResult()`.

Studying the processes of stubborn permission requests, we find that method `onRequestPermissionsResult()` could be considered as the beginning of analyzing stubborn behaviors. Besides, we find that some app do not get the response of the permission request in the recommended method `onRequestPermissionsResult()`. There are other ways of indicating whether users grant the requested permission or not. First, an app could call `checkSelfPermission()` to check permissions and perform stubborn permission request if it is not granted. Hence, method `checkSelfPermission()` and `checkCallingorSelfPermission()` are also considered as entry points by our approach. Second, an app could indicate whether it has certain permissions. For example, an app could invoke an Android API (*e.g.*, `getLongitude()`) which requires a dangerous permission and utilize Java exception handling to infer whether the app has the permission. Table 1 lists some sensitive APIs and their required corresponding permissions. Hence,

Table 1: Sensitive APIs and the corresponding permissions

Method	Permission
com.android.internal.telephony.cdma.CDMAPhone: java.lang.String getDeviceId()	READ_PHONE_STATE
android.location.Location:double getLongitude()	ACCESS_COARSE_LOCATION, ACCESS_FINE_LOCATION
com.android.nfc.NfcService:void onSeApduReceived(byte[])	NFC
com.android.server.sip.SipService:void open(android.net.sip.SipProfile)	USE_SIP
android.media.AudioManager: void setRingerMode(int)	RECORD_AUDIO
android.telephony.SmsManager:void sendTextMessage(...)	SEND_SMS

the try-catch blocks which invoke sensitive APIs are also considered as entry points.

Our approach locates all the entry point in an app, by searching the method names of entry point in the decompiled Jimple codes. Particularly, method `onRequestPermissionsResult()` is a callback method which would be invoked by Android system and it is overridden by the app in order to handle the permission request response. As a result, this method is found by retrieving all the third party classes of the app. For the try-catch blocks, we search all the try blocks in the Jimple codes for sensitive APIs (list from PScout [2]). If found, we record the corresponding catch blocks as entry points for further analyzing.

Particularly, we denote *exit-methods* as APIs that would finish the activities or cause the app to exit. Table 2 lists the exit-methods monitored by our analysis. Some of the methods directly finish the activities or terminate the app process. For example, method `finish()` and `finishActivity()` finish an Android activity. The `System.exit()` method quits the current program by terminating the running Java virtual machine. Method `killBackgroundProcesses()` kills all background processes associated with the given package. Method `killProcess()` kills the process with the given PID. `finishAndRemoveTask()` finishes all activities in this task and removes it from the recent tasks list. Another way of stopping users from utilizing the functionalities of an app is to hide the activities of the app to background. `moveTaskToBack()` moves the task containing this activity to the back of the activity stack.

Table 2: The exit-methods

Class	API
android.app.Activity	finish
android.support.v4.app.ActivityCompat	finishAffinity
java.lang.System	exit
android.app.ActivityManager	killBackgroundProcesses
android.os.Process	killProcess
android.app.Activity	moveTaskToBack
android.app.ActivityManager.AppTask	finishAndRemoveTask

Algorithm 1 Identifying stubborn permission requests

1: **Input:** the entry point methods *entryPoints*, and the APK file *app.apk*
2: **Output:** whether the app has the stubborn permission request behavior
3: Begin
4: *callgraph* ←generateCHACallgraphWithSoot(*app.apk*)
5: **for all** $ep \in entryPoints$ **do**
6: **if** reachabilityAnalysis($ep, callgraph$) **then**
7: **return true**
8: **end if**
9: **end for**
10: **return false**
11: End

3.2.2 Finding Stubborn Permission Requests

To figure out whether the entry points would lead to the exit of an app under certain circumstance, we traverse the call graphs which are rooted from the entry points. Here, we introduce our reachability analysis, which finds whether an entry point method would directly or indirectly invoke methods that finish activities or cause the app to exit.

Our reachability analysis is implemented by finding method invocation chains from the entry points to exit-methods in an app. The method invocation chain indicates that checking the permission request response would finally lead to the exit of an app. In particular, the invocation chain is a path in the call graph which starts with an entry point and ends with an exit-method. Hence, if there is a method invocation chain from an entry point method to an exit-method, we report that the app contains stubborn permission requests.

Algorithm 1 shows how our approach finds stubborn permission requests. It accepts the entry points and the apk file as inputs. The output of this algorithm is a judgment about whether this app contains stubborn permission requests. Generally, Algorithm 1 recursively searches the methods that could be triggered from the entry point

Algorithm 2 reachabilityAnalysis

```
1: Input: a method ep, a call graph callgraph, the exitMethods
2: Output: whether method ep has the stubborn permission request behavior
3: Begin
4: subgraph ← callgraph.rootedWith(ep)
5: children ← subgraph.getChildren(ep)
6: for all child ∈ children do
7:   if child ∈ third party packages then
8:     if reachabilityAnalysis(child, subgraph) then
9:       return true
10:    end if
11:  else
12:    if child ∈ exitMethods then
13:      return true
14:    end if
15:  end if
16: end for
17: return false
18: End
```

methods.

In detail, Algorithm 1 first decompiles the apk file into Jimple codes, then it utilizes Soot to generate a call graph using the class hierarchy analysis (CHA) (*i.e.*, line 4), which conservatively estimates possible receivers of dynamically-dispatched messages. As a result, for example, a *virtual method* could be invoked, which is due to the use of polymorphism through which it is possible for a subtype to override methods defined in its super-types. The actual target of the virtual call is determined at run time. The generated call graph would list the possible target method in it.

Then we apply the reachability analysis (Algorithm 2) for each entry point (*i.e.*, line 5-9 in Algorithm 1). In the reachability analysis, Algorithm 2 first gets a subgraph rooted with the method *ep* from the the call graph of the app. Then, it travels the subgraph to check each method. If a method in the call graph belongs to the third party libraries (*i.e.*, defined by the app), this algorithm would recursively search for exit-methods inside the method (*i.e.*, line 6 to line 16 in Algorithm 2). If one of the exit-methods is found in the sub-callgraphs rooted with an entry point method, the algorithm would report a recognition of stubborn permission request.

4 Evaluation

4.1 Experimental Setup

All experiments were conducted on a 4-processor 16GB-RAM machine, and all the apps analyzed in this section were collected from four third party Android app markets in China (*i.e.*, wandoujia, anzhi, baidu-shouji, tecentyingyongbao) and Google Play. The collected apps were among the top popular apps in each category sorted by the app markets.

4.2 Real-world Apps Study

We first analyzed 104 apps downloaded from the four third party Android app markets to figure out how common is the stubborn permission request in real-world apps of China. The selected apps were the most popular apps (sorted by the downloads) of all the apps in the market, and all of them were listed by the four app markets. Note that some apps may have different release versions of apps targeted for different kinds of mobile devices. Besides, some of the apps may have customized versions cooperated with the app market for the purpose of advertising, but main functionalities of these apps are the same. In our experiments, we found that different versions of an app have the same behaviors of permission requesting. Hence, we treated the apps with same name as the same app. As a result, we found that 25 (*i.e.*, 24%) tested China apps contained stubborn permission request behaviors.

We also investigated corresponding apps from Google Play. Google Play serves as the official app store for the Android operating system. Only 37 out of the 104 Chinese apps are listed in Google Play. Other apps do not provide international services. Among the 37 apps, 4 (*i.e.*, 11%) apps contain stubborn permission requests. We can conclude that apps in Google Play may also contain stubborn permission requests, but apps in Google Play are with lower ratio of containing stubborn permission requests than apps in third party markets of China. Based on the above experimental results, we suggest that app markets pay more attention to stubborn permission requests in apps, especially third party Android app markets.

Table 3: Results of market apps study (P: precision, R: recall)

Category	analyzed	stubborn	detected	P	R
Movie &Music	19	2	2	100%	100%
News	20	7	7	100%	100%
Social	18	4	3	100%	75%
Tools	19	6	6	100%	100%
Sport	19	5	4	100%	80%
Shopping	20	7	6	100%	86%
Weather	20	6	6	100%	100%

Furthermore, in order to investigate stubborn permission requests in different categories of apps, we analyzed 140 apps from 7 different categories, each of which contains 20 apps, and each app was analyzed for maximum 30 minutes. As apps providing different functionalities require different permissions, the occurrence frequency of stubborn permission requests may be different. For example, apps which provide "map" function naturally requires `GPS` permission, and the stubborn permis-

sion requests are more likely to be found in these apps. Table 3 lists the results of our investigation. Of the 140 apps, 5 (3.5%) apps were not analyzable, which is caused by exceeding the RAM limit or the 30 minutes timeout, and Soot exceptions while transforming bytecode to Jimple representation.

The results show that stubborn permission requests occur differently in each app category. Particularly, the stubborn permission request occurs more among apps in the categories of *news*, *tools*, *shopping* and *weather* than others, which indicates that app providers in these categories are more likely to collect users' privacy. Apps in these categories provide functionalities frequently used by users, which increases the possibility of leaking privacy. As a result, we suggest app providers of these apps regulate the behaviors in the apps of requesting dangerous permissions and utilizing users' private data.

Table 4: Results of popular apps from different app markets

App Name	Wandoujia	Anzhi	Baidu-shouji	Tecent-yingy-ongbao	Google Play
UC browser	◯	◯	◯	◯	×
WeChat	◯	◯	◯	◯	◯
Taobao	◯	◯	◯	◯	◯
Zhihu	×	×	×	×	×
Meituan	◯	◯	◯	◯	◯
DiDi	◯	◯	◯	◯	◯
Ctrip	×	×	×	×	×
zhifubao	◯	◯	◯	◯	◯
Pinduoduo	×	×	×	×	–
QQ Music	◯	◯	◯	◯	–
Toutiao	×	×	×	×	–
Facebook	–	–	–	–	×
Reddit	–	–	–	–	×
Quora	×	×	×	×	×

◯ means that the app contains stubborn permission requests.
× means the app does not contain stubborn permission requests.
– means that the app is not listed on the corresponding app market.

Next, we present some detection results of popular apps from different app markets. Table 4 lists our detection results. The improper behavior of stubbornly permission requests is not rare in the most popular apps. We observe different results regarding to different app markets. For example, some apps from Chinese app markets contain stubborn permission requests, but the releases of them on Google Play does not contain stubborn permission requests (*e.g.*, *UC browser*). Some apps contain stubborn permission requests, and these apps are only listed on Chinese app markets (*e.g.*, *QQ Music*). We also observe different behaviors that do not contain stubborn permission requests. For example, *Zhihu* does not request permission when it is started to run, *Ctrip* requests permissions but it does not exit even if users refuse the request.

4.3 Precision and Recall

To get the precision and recall of our analysis results, we manually installed the tested apps and checked whether these apps have stubborn permission requests. For apps which require users to sign up the app before enjoying the functionalities, we created accounts to test these apps. We ran each app for 5 minutes, manually triggering all the UI elements of each app's activities, until we found a stubborn permission request. Finally, the ground truth of whether the tested apps have stubborn permission requests was collected.

We consider two evaluation metrics, the precision and recall.

Precision: The fraction of permission requests correctly identified as stubborn among those reported by our static approach.

Recall: The fraction of permission requests correctly identified as stubborn among those manually tested, *i.e.*, the ground truth.

Given the ground truth information and the detection results, there are four possible outcomes: True positive(TP), true negative (TN), false positive (FP) and false negative (FN). TP means that an app contains improper behavior of stubbornly requesting sensitive permissions with respect to the ground truth and it is detected by our approach. TN means that an app does not contain improper behavior of stubbornly requesting sensitive permissions with respect to the ground truth and our approach does not find stubborn permission requests in the app. FP means that an app does not contain improper behavior of stubbornly requesting sensitive permissions with respect to the ground truth but our approach reports that the app contains stubborn permission behaviors. FN means that an app contains improper behavior of stubbornly requesting sensitive permissions with respect to the ground truth but our approach does not find stubborn permission requests in the app. Finally, the precision and recall are computed by the following formulas:

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

The precisions and the recalls of the analysis results for apps from different categories are listed in Table 3 (the last two columns). The third column lists the manually analyzed results. For all the app categories, we got 100% detection precision, which means that all the reported apps in our detection results have stubborn permission requests. The recalls of our detection are also high, which indicates the effectiveness of our work. There exist apps which actually have stubborn permission requests, but our static analysis approach didn't recognize them. Two reasons account for this situation. First, there are other covert ways of checking whether an app has certain dangerous permission. For example, some malicious app may use Java reflection to obfuscate method calls. It

is difficult for static analysis based approaches to handle reflection [26]. In this situation, the method calls of entry point methods may be missed by our static approach for that our approach does not support handling reflection currently. Second, we find that there exist apps which do not exit after the permission requests are denied by users. Instead, these apps continuously request for permissions. Even if the standard dialog of requesting permissions would not be shown after users choose "Don't ask again", these app would present their own dialog (*e.g.*, as shown in Figure 1b and Figure 1c). Users could finish the app by clicking the default Android "home" button. We leave these as future work.

We believe that our approach is effective in detecting improper behaviors of stubbornly requesting permissions in Android applications. Although manual detection can achieve high precision and recall, it is not applicable for massive app audition. As our static analysis approach can achieve similar detection results compared with manual detection and it does not require human assist, we believe it can help app markets or analysts to automatically detect stubborn permission requests in Android apps.

4.4 Findings

During our analysis on the market apps, we have some findings. The `PHONE` and `STORAGE` are the two most frequently requested permissions and they are also the top permissions occurring in the stubborn permission requests. We studied the corresponding apps that stubbornly request the two permissions to figure out why these apps insist in getting the permissions.

We find that the some of the Android permission groups are not properly designed , which actually contributes to the concern about privacy leakage of stubborn permission requests. For example, the `PHONE` permission group contains `READ_PHONE_STATE`, `CALL_PHONE`, `READ_CALL_LOG`, `WRITE_CALL_LOG`, *etc.* Some apps need to read the IMEI (International Mobile Equipment Identity) of a device for identifying the unique phone. For instances, in some voting systems, it is required that each device only has one vote. Mobile app developers need to understand who are using their apps, and the IMEI is often used to distinguish different users [22]. Hence, these apps request for the corresponding permission: `READ_PHONE_STATE`. However, the system informs users by the permission group `PHONE`. Users may be worried about granting the app this group of permissions for that if the app requests for other permissions in this group latter, the system would directly grant it the permissions without informing users again. The permissions in the permissions group are organized by Android permissions mechanism with the same sensitive resource, but as the situation in this case, the `READ_PHONE_STATE` is more frequently used by apps than other permissions in its group. We believe that the frequency of the permissions used by apps should also be taken into consideration. Based on our study, we suggest that this permission could be taken out from the permission group `PHONE` in order to reduce the privacy concern.

We also find that different versions of an app may have different behaviors of permission requests. For example, *Weibo* and *Weibo international* are different versions of the client app of *Sina Weibo*. The *Weibo* is an interface for Chinese market, and the *Weibo international* aims to be adapted by other cultures. However, *Weibo* contains stubborn permission requests, while *Weibo international* does not have ones. Different markets may have different data protection rules, which may account for this situation that two apps of the same company have different behaviors of stubborn permission requests. We believe that users' privacy should be put it in the first place, and app providers should follow the same data protection rules to develop their apps.

We observe that all the stubborn permission requests occur at the initialization of an app. The purpose of the stubborn permission is to urge users to grant the requested permissions. Hence, stubborn permission requests at the initialization of an app put pressure on users: if they do not grant the permissions, they would not enjoy normal app functionalities. As a result, it increases the possibility for the app to get the requested permissions. Besides, we observe some apps, which contain stubborn permission requests, do not check whether they have the permissions at runtime in the codes. Hence, stubbornly requesting permissions at the initialization of an app guarantees that the app always has the requested permissions at runtime, which reduces the workload of the app developers. Based on these observations, we suggest the analyzers and users pay more attention to the apps which stubbornly request permissions at initialization.

5 Related Work

Prior work demonstrates that install-time prompts of requesting permissions fail to protect users' privacy because users do not comprehend these permission requests or pay attention to them [12, 17]. Users often do not understand which permission correspond to which functionalities in apps before they are familiar with the apps. As a result, users are prone to grant the permissions. Apex [23] and πbox [19] provide users with the ability to grant permissions to the app at runtime. This feature is now integrated in Android since the version of Marshmallow. In our work, the stubborn permission request occurs at the initialization of Android apps, which has the same problem that users are not familiar with apps. Moreover, stubbornly requesting permission urges users to grant the permissions.

Researches have designed systems to recommend permissions for app developers to properly request permissions [3, 16]. These researches are based on mining technique or collaborative filtering technique. Other researchers have developed systems to to predict permission decisions at runtime based on contextual information and

machine learning methods [24]. By requiring users to report privacy preferences, clustering algorithms have been used to define user privacy profiles even in the face of diverse preferences [21].

Wijesekera *et al.* [28] build a classifier to make privacy decisions on the user's behalf by detecting when context has changed and, when necessary, inferring privacy preferences based on the user's past decisions and behavior. It automatically grants appropriate resource requests without further user intervention, denies inappropriate requests, and only prompts the user when the system is uncertain of the user's preferences.

There is a large body of work researching the improper use of permissions in Android permissions. Wei *et al.* [27] find that some Android Applications do not follow the principle of least privilege, intentionally or unintentionally requesting permissions which are not related to the declared app functions. Fauzia *et al.* investigate the combined effects of permissions and intent filters to distinguish between the malware and benign apps [15]. Qian *et al.* [25] use static analysis to determine whether an app has potential risks, and then embed monitoring Smali code for sensitive APIs.

As a result, their approach could reveal the malicious behaviors of applications leaking users' private data. Zhao *et al.* [30] extract the API packages, risky API functions and permission information and then use convolutional neural network to identify Android malwares. Pegasus [6] focuses on detecting the malicious behavior that can be characterized by the temporal order in which an app uses APIs and permissions. It can automatically detect sensitive operations being performed without the user's consent. Our work detects a kind of improper behavior that stubbornly requests dangerous permissions.

To enhance the Android permission mechanism, MockDroid [4] allows users to mock an application's access to a resource. It offers users with binary options that either revoking access to particular resources or providing full access to the app. However, MockDroid only works for explicitly requested resources. To deal with innocuous sensors, IpShield [5] performs monitoring of every sensor accessed by an app and allows users to configure privacy rules which consist of binary privacy actions on individual sensors. Blue Seal [13] extends the Android permission mechanism with semantic information based on information flows, which allows users to examine and grant information flows within or across multiple applications. It can remove unnecessary permissions from over privileged apps and synthesize flow permissions for the app. FineDroid [29] associates each permission request with its application context and provides a fine-grained permission control. FineDroid also features a policy framework to flexibly regulate context-sensitive permission rules. SmarPer [24] relies on contextual information and machine learning methods to predict permission decisions at runtime.

6 Conclusion

This paper presents a static analysis approach which targets for detecting stubborn permission requests: if users do not grant the required dangerous permissions, the app would not provide any functionalities. This stubborn behavior threatens users' privacy for that if users yield to it, the app would get full access to sensitive data and the users are not aware of how the app would use the sensitive data. By statically analyzing the decompiled codes, we identify the stubborn permission requests. Our experimental results indicate that our approach is effective in detecting the stubborn requests. Our work could be utilized by app markets so that users can be informed if an app contains stubborn permission requests.

Acknowledgments

The research is supported by National Natural Science Foundation of China under Grant No.61572453, No.61202404, No.61520106007, No.61170233, No.61232018, No.61572454, Natural Science in Colleges and Universities in Anhui Province under Grant No.KJ2015A257, and Anhui Provincial Natural Science Foundation under Grant No.1508085SQF215. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. L. Traon, D. Octeau, and P. McDaniel, "Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps," in *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, vol. 49, no. 6, pp. 259–269, 2014.

[2] K. W. Y. Au, Y. Zhou, Z. Huang, and D. Lie, "Pscout: analyzing the android permission specification," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pp. 217–228, 2012.

[3] L. Bao, D. Lo, X. Xia, and S. Li, "What permissions should this android app request?," in *International Conference on Software Analysis, Testing and Evolution (SATE)*, pp. 36–41, 2016.

[4] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan, "Mockdroid: Trading privacy for application functionality on smartphones," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, pp. 49–54, 2011.

[5] S. Chakraborty, C. Shen, K. R. Raghavan, Y. Shoukry, M. Millar, and M. Srivastava, "Ipshield: A framework for enforcing context-aware privacy," in *11th USENIX Symposium on Networked Systems*

Design and Implementation (NSDI'14), pp. 143–156, 2014.

[6] K. Z. Chen, N. M. Johnson, V. D'Silva, S. Dai, K. MacNamara, T. R. Magrino, E. X. Wu, M. Rinard, and D. X. Song, "Contextual policy enforcement in android applications with permission event graphs," in *NDSS*, 2013. (https://www.cs.cornell.edu/~tmagrino/papers/ndss13-pegasus.pdf)

[7] Developer.android.com., *Requesting Permissions at Run Time*. (http://developer.android.com/training/permissions/requesting.html)

[8] Developer.android.com., *Permissions Overview*. (https://developer.android.com/guide/topics/permissions/overview.html#normal-dangerous)

[9] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, M. Conti, and M. Rajarajan, "Android security: A survey of issues, malware penetration, and defenses," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 998–1022, 2015.

[10] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, pp. 627–638, 2011.

[11] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the eighth symposium on usable privacy and security*, pp. 3, 2012.

[12] A. Gorla, I. Tavecchia, F. Gross, and A. Zeller, "Checking app behavior against app descriptions," in *Proceedings of the 36th International Conference on Software Engineering*, pp. 1025–1035, 2014.

[13] S. Holavanalli, D. Manuel, V. Nanjundaswamy, B. Rosenberg, F. Shen, S. Y. Ko, and L. Ziarek, "Flow permissions for android," in *IEEE/ACM 28th International Conference on Automated Software Engineering (ASE'13)*, pp. 652–657, 2013.

[14] IDC, *Smartphone OS Market Share*, 2018. (http://www.idc.com/promo/smartphone-market-share/os).

[15] F. Idrees and M. Rajarajan, "Investigating the android intents and permissions for malware detection," in *IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'14)*, pp. 354–358, 2014.

[16] M. Y. Karim, H. Kagdi, and M. Di Penta, "Mining android apps to recommend permissions," in *IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER'16)*, vol. 1, pp. 427–437, 2016.

[17] P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall, "A conundrum of permissions: Installing applications on an android smartphone," in *International Conference on Financial Cryptography and Data Security*, pp. 68–79, 2012.

[18] P. Lam, E. Bodden, O. Lhoták, and L. Hendren, "The soot framework for java program analysis: A retrospective," in *Cetus Users and Compiler Infastructure Workshop (CETUS'11)*, vol. 15, pp. 35, 2011.

[19] S. Lee, E. L. Wong, D. Goel, M. Dahlin, and V. Shmatikov, "πbox: A platform for privacy-preserving apps," in *NSDI*, pp. 501–514, 2013.

[20] L. Li, A. Bartel, T. F. Bissyande, J. Klein, Y. Le Traon, S. Arzt, S. Rasthofer, E. Bodden, D. Octeau, and P. McDaniel, "IccTA: Detecting inter-component privacy leaks in android apps," in *IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE'15)*, 2015. ISBN: 978-1-4799-1934-5.

[21] J. Lin, B. Liu, N. Sadeh, and J. I. Hong, *Modeling Users' Mobile App Privacy Preferences: Restoring Usability in a Sea of Permission Settings*, 2014. (https://www.usenix.org/system/files/conference/soups2014/soups14-paper-lin.pdf)

[22] W. Liu, Y. Zhang, Z. Li, and H. Duan, "What you see isn't always what you get: A measurement study of usage fraud on android apps," in *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 23–32, 2016.

[23] M. Nauman, S. Khan, and X. Zhang, "Apex: extending android permission model and enforcement with user-defined runtime constraints," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 328–332, 2010.

[24] K. Olejnik, I. Dacosta, J. S. Machado, K. Huguenin, M. E. Khan, and J. Hubaux, "Smarper: Context-aware and automatic runtime-permissions for mobile devices," in *IEEE Symposium on Security and Privacy (SP'17)*, pp. 1058–1076, 2017.

[25] Q. Qian, J. Cai, M. Xie, and R. Zhang, "Malicious behavior analysis for android applications," *International Journal of Network Security*, vol. 18, no. 1, pp. 182–192, 2016.

[26] S. Rasthofer, S. Arzt, M. Miltenberger, and E. Bodden, "Harvesting runtime values in android applications that feature anti-analysis techniques," in *Proceedings of the Annual Symposium on Network and Distributed System Security (NDSS'16)*, 2016. (https://www.bodden.de/pubs/ssme16harvesting.pdf)

[27] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos, "Permission evolution in the android ecosystem," in *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 31–40, 2012.

[28] P. Wijesekera, A. Baokar, L. Tsai, J. Reardon, S. Egelman, D. Wagner, and K. Beznosov, "The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences," in *IEEE Symposium on Security and Privacy (SP'17)*, pp. 1077–1093, 2017.

[29] Y. Zhang, M. Yang, G. Gu, and H. Chen, "Rethinking permission enforcement mechanism on mobile

systems," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2227–2240, 2016.

[30] Y. Zhao and Q. Qian, "Android malware identification through visual exploration of disassembly files," *International Journal of Network Security*, vol. 20, no. 6, pp. 1061–1073, 2018.

Biography

Jianmeng Huang received the B.S. degree in computer science from University of Science and Technology of China in 2013. He is currently working towards the Ph.D. degree at the Department of Computer Science and Technology, University of Science and Technology of China. His current research interests include information security and mobile computing.

Wenchao Huang received the B.S. and Ph.D degrees in computer science from University of Science and Technology of China in 2006 and 2011, respectively. He is an associate professor in School of Computer Science and Technology, University of Science and Technology of China. His current research interests include information security, trusted computing, formal methods and mobile computing.

Fuyou Miao received his Ph.D of computer science from University of Science and Technology of China in 2003. He is an associate professor in the School of Computer Science and Technology, University of Science and Technology of China. His research interests include applied cryptography, trusted computing and mobile computing.

Yan Xiong received the B.S., M.S., and Ph.D degrees from University of Science and Technology of China in 1983, 1986 and 1990 respectively. He is a professor in School of Computer Science and Technology, University of Science and Technology of China. His main research interests include distributed processing, mobile computing, computer network and information security.

On the Security of a Practical Constant-Size Ring Signature Scheme

Jianhong Zhang[1], Wenle Bai[1,2], and Zhengtao Jiang[1,3,4]
(Corresponding author: Weile Bai)
School of Electronic and Information Engineering, North China University of Technology[1]
Beijing 100144, China
(Email: bwl@ncut.edu.cn)
Guangxi Key Laboratory of Cryptography and Information Security[2]
Guangxi Key Lab of Multi-source Information Mining & Security[3]
School of Electronic and Information Engineering, China Communication University[4]
(Received June 15, 2018; Revised and Accepted Nov. 2, 2018; First Online July 8, 2019)

Abstract

Due to decentration and anonymity, "bitcoin" cryptocurrency is widely paid attention. However, it only furnishes pseudo-anonymity instead of authentic anonymity. As an anonymous technique, ring signature is a candidate to provide authentic anonymity in cryptocurrency [2]. However, in most of existing ring signatures, the length of signature grows linearly with the size of the ring. To construct constant-size signature, Qin *et al.* recently proposed a practical constant-size ring signature scheme, and claimed that their scheme can provide unforgeability and anonymity which are two basic security requirments of ring signature. Unfortunately, in this letter, we show that their scheme is insecure against unforgeability attack and anonymity attack. Finally, the two detail attacks are given.

Keywords: Anonymity; Constant-Size Ring Signature; Security Attack; Unforgeability

1 Introduction

As a decentralized distributed public ledger, blockchain can furnish trust to operations between unrelated parties, without requiring the collaboration of a trusted third party. However, the public verifiability and decentration of blockchain transaction often do not provide the strong security and privacy properties required by the users. It can only provide pseudonymity instead of true anonymity. In cryptocurrency, ring signature, stealth address, and zero-knowledge proof are several cryptographic techniques which can achieve privacy protection for transaction entities.

In particular, ring signature can often been used to protect the identity anonymity of the user. To provide real anonymity of transaction party, ring signature is applied to conceal the origin of a transaction in the 'Monero' cryptocurrency. For the perspective of the outsider, it can not distinguish who is the actual sponsor of transaction.

In 2001, a novel anonymous signature conception [14] was invented by Rivest, Shamir and Tauman, it was named as **ring signature** since the structure of signature generation looks like a ring. Like group signature, a user is allowed to produce a signature on behalf of the whole group without the cooperation of the other users of the group in ring signature. Anonymity and unforgeability [1, 3, 7, 9, 11] are two basic properties of a secure ring signature. Anonymity can guarantee the identity privacy of the actual signer. Unforgeability can ensure the security of signature algorithm since it can prevent an adversary from forging a signature of new message m. These properties make that ring signature has very important application such as anonymous authentication in VANET. However, most of the existing ring signature schemes exist a common flaw: "*The length of ring signature is linear with the size of the ring.*". Therefore, the larger the ring size is, the longer the length of ring signature is.

To reduce the length of ring signature, Chandran *et al.* presented the first sub-linear length ring signature [5] in the standard model by utilizing private information retrieval technique. However, the security of their scheme builds on the composite-order bilinear group which is inefficient. Later on, Ghadafi constructed a sub-linear size ring signature in the prime-order setting [8].

There only exist two constant-size ring signature schemes so far. One is Dodis *et al.*'s anonymous identification scheme [6] which is based on strong RSA problem, the other is Bose *et al.*'s ring signature [4] which is based on two Diffie-Hellman assumption. Nevertheless, the two scheme is inefficient in practice since the scheme in [6] makes use of the strong RSA based instantiation which uses quite complex Σ- protocols, and the scheme in [4] is based on the q-strong Diffie- Hellman

Table 1: Notions

Notions	Implication
$\mathbb{G}$,$\mathbb{G}_T$:	two groups with the same order q
e :	A bilinear pairing map
PPT:	Probability polynomial time
DLP:	discrete logarithm problem
ACC :	A dynamic accumulator
PPT:	the probabilistic polynomial time
H_0, H_1:	two cryptographic hash functions
(x_i, PK_i):	public-private pair of user i

(q-SDH) assumption and the symmetric external Diffie Hellman (SXDH) assumption via the Boneh- Boyen signature scheme. These techniques make the two schemes less efficient in practice.

Our contributions: Recently, based on the discrete logarithm problem (for short, DLP) assumption, Qin *et al.* proposed a practical constant-size ring signature scheme [17]. Their construction is very simple and efficient. And they claimed that their scheme can provide security proof of unforgeability and anonymity. Unfortunately, in this work, we find that their scheme is insecure by analyzing the security of Qin et al.'s scheme. Their scheme suffers from two security flaws. One is that it exists universal forgeability, namely, any one can forge a signature on arbitrary a message; The other is that it can not achieve anonymous prevention of real signer's identity. Finally, the detail attacks are given.

2 Preliminaries

In this work, we make use of bilinear map technique to construct our scheme. To make our paper self-contained, we will introduce some necessary cryptographic background which is related to bilinear map . And they are also the basis of achieving the security of our proposed scheme.

2.1 Bilinear Maps [10, 15, 16]

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of the same prime order q, g be a generator of group $\mathbb{G}_1$. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a map that for all $g, h \in \mathbb{G}_1$ and $a, b \in_R Z_q^*$, $e(g^a, h^b) = e(g, h)^{ab}$. And there exists a computable algorithm that can efficiently compute e and $e(g, g) \neq 1$.

2.2 Accumulator

Definition 1. *An accumulator is a tuple (X_χ, F_χ), where $\chi \in N$, N is the set of positive integers. X_χ is called the value domain of the accumulator, and F_χ is a collection of pairs of functions such that each $(f, y) \in F_\chi$ is defined as $f : U_f \times X_f \to U_f$ for some $X_\chi \subseteq X_f^{ext}$, and $y : U_f \to U_y$ is a bijective function where U_f and U_y denote the value domain of functions f and y respectively. In addition, an accumulator should satisfy the following properties.*

1) Efficient generation. There exists an efficient algorithm that takes as input a security parameter χ and outputs a random element $(f, y) \in_R F_\chi$, possibly together with some auxiliary information. And in the following sections, we denote the algorithm by ACC.Gen.

2) Quasi-commutativity. For every $\chi \in N$, $(f, y) \in F_\chi, u \in U_f$, $x_1, x_2 \in X_\chi$: $f(f(u, x_1), x_2) = f(f(u, x_2), x_1)$. For any $\chi \in N$, $(f, y) \in F_\chi$ and $X = \{x_1, x_2, \cdots, x_n\} \subset X_\chi$, we call $y(f(f(u, x_1), \cdots, x_n))$ the accumulated value of the set X over u. Due to quasi-commutativity, the value $y(f(f(u, x_1), \cdots, x_n))$ is independent of the order of x_i and is denoted by $f(u, X)$.

3) Efficient Evaluation. For every $(f, y) \in F_\chi, u \in U_f$ and $X \subset X_\chi$ with polynomially-bound size: $y(f(u, X))$ is computable in time polynomial in χ, and we use ACC.Eval to represent the process of computing the accumulated value. Also, there is a witness w meaning that some variable x has been accumulated within $v = f(u, X)$ iff $f(w, x) = v$, and we use ACC.Wit to denote computing the witness w.

For simplicity, in the context, we adopt the accumulator in [13]. Namely, the functions (f, y) is defined as follows:

$$f : (\beta, x) \to \beta(x + d), y : x \to x$$

where $d \in Z_q$ and $\beta \in Z_q$, and $y(x) = x$ is an identity function. For this accumulator, we have $f(\beta, X) = \beta(x_1 + d) \cdots (x_n + d)$ for a set $X = \{x_1, x_2, \cdots, x_n\}$.

Obviously, all the above three properties in Definition 1 are satisfied.

2.3 Ring Signature Definition

A ring signature scheme is a tuple (Setup, Gen, Sign, Verify) of PPT algorithms, where each of them means generating a key pair, signing a message, and verifying the signature for the message using the corresponding public keys, respectively. Formally they are described as follows.

1) Setup(1^l). It takes as input a security parameter l, and outputs the system parameters *params*.

2) Gen(*params*, i). It inputs the identity information i of a user and *params*, and outputs a public key PK_i and a private key SK_i for each member i.

3) $Sign(SK, m, R)$. The signer outputs a signature δ on a message m with respect to a ring R using the signing key SK_i. We assume that the number of public keys in the ring $|R| > 2$, and there is exactly one public key in R corresponding to the signing key SK_i, and all the keys in R are generated by Gen.

4) Verify(δ, m, R). The verifier can be anyone, including the adversary, who verifies the signature on a message m with respect to the ring R. If the verifier accepts the signature, then the algorithm returns 1; otherwise, returns 0.

3 Reviews of Qin *et al.*'s Constant-Size Ring Signature

To achieve constant-length ring signature, Qin *et al.* proposed a practical constant-size ring signature (for short, PCRS) in [17]. The PCRS scheme consists of four algorithms. We will briefly review their algorithms. Please the interesting readers refer to [17] for the detail. For convenience, the used notations in the following parts are summarized in Table 1.

3.1 System Initialization

Taking a security parameter λ as input, it outputs two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_T$ with the same prime order p, and the discrete logarithm problem in $\mathbb{G}_1$ is hard. Let g be a generator of group $\mathbb{G}_1$ and $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ be a bilinear pairing map. H_0 and H_1 are hash functions which satisfy $H_0 : \{0,1\}^* \to Z_q$ and $H_1 : \{0,1\}^* \to Z_q$, where q is also a prime which satisfies $q|p-1$. For each user $i \in \{1, \cdots, n\}$, it chooses x_i as its private key and calculates public key $PK_i = g^{x_i}$. Let R denote the public key list of n users,namely $R = \{PK_1, \cdots, PK_n\}$. Let AAC denote an accumulator, refer to [17] for the detail. And then we invoke AAC.Gen to produce $f : (\beta, x) \to \beta(x+d)$ and the corresponding $P_{pub} = g^d$. Finally, the system parameter $Param = \{\mathbb{G}_1, \mathbb{G}_T, e, p, q, g, f, P_{pub}, H_0, H_1\}$.

3.2 Signing

Let $L = \{H_0(PK_i)\}_{i=1}^n$ and $L' = \{H_0(PK_i)\}_{i=1, i\neq s}^n$. Given a public key list $R = \{PK_1, PK_2, \cdots, PK_n\}$ and system parameter $Param$, to produce a signature on message m, a user s with public-private key pair (x_s, PK_s) executes the following steps:

1) First, it calculates $V = ACC.Eval(f, L)$ and $W = ACC.Wit(f, L')$. Note that $V = W(H_0(PK_s) + d)$.

2) Next, it uniformly samples $r \in Z_q$ to calculate $U = W + r$ and $P_U = g^U$.

3) Pick $k_1, k_2 \in Z_q$ at random to calculate $\prod = e(g, P_U)^{-k_1} e(P_{pub}, g)^{k_2} e(R', g)$ where $e(R', g) = \prod_{i=1, i\neq s}^n e(PK_i, g)$.

4) And it calculates

$$c = H_1(m||V||P_{pub}||P_U||\prod||R)$$

where $R = \{PK_1, \cdots, PK_n\}$.

5) Finally, it computes

$$\begin{aligned} s_1 &= k_1 + cH_0(PK_s) \\ s_2 &= k_2 + cr \\ s_3 &= crH_0(PK_s) - x_s. \end{aligned}$$

6) The resultant ring signature is $\delta = (c, P_{pub}, P_U, V, s_1, s_2, s_3)$

3.3 Verifying

Given a message m and its ring signature $\delta = (c, P_{pub}, P_U, V, s_1, s_2, s_3)$ as well as public key list R, a verifier can conduct the following procedure. First, it calculates

$$\begin{aligned} \Pi' &= e(g, P_U)^{-s_1} e(P_{pub}, g)^{s_2} e(g, g)^{s_3} e(P_{pub}, P_U)^{-c} \\ &\quad \cdot e(g, g^V)^c e(R, g). \end{aligned}$$

And then it checks whether $c \stackrel{?}{=} H_1(m||V||P_{pub}||P_U||\Pi'||R)$. If it holds, then the signature is accepted; otherwise, refuse it.

4 Security Analysis

Recently, based on the DLP problem, Qin *et al.* proposed a PCRS scheme. Their scheme is more efficient than the existing two constant size ring signature schemes in terms of computational cost. At the same time, they also claim that their scheme can achieve anonymity of the signer's identity and provide perfect zero knowledge for the verifier. By analyzing the security of their scheme, we find that their scheme does not achieve anonymity. Given a ring signature δ, a verifier can know which signer produce the signature δ. Furthermore, given two ring signatures δ and δ', the verifier can know whether the two ring signatures are from the same signer. The detail attacks are given as blow.

4.1 Attack on Anonymity

Given a ring signature $\delta = (c, P_{pub}, P_U, V, s_1, s_2, s_3)$ on message M, an attack A first computes

$$\begin{aligned} \Pi &= e(g, P_U)^{-s_1} e(P_{pub}, g)^{s_2} e(g, g)^{s_3} \\ &\quad \cdot e(P_{pub}, P_U)^{-c} e(g, g^V)^c e(R, g) \\ c &= H_1(m||V||P_{pub}||P_U||\Pi') \end{aligned}$$

For $j = 1$ to n
{

1) It computes $\bar{k}_1 = s_1 - c \cdot H_0(PK_j)$;

2) And then it computes $s_2 H_0(PK_j) - s_3 = k_2 H_0(PK_j) + x_s$;

3) And it checks

$$\begin{aligned} & e(P_{pub}, PK_j) \cdot \left(\frac{\Pi' \cdot e(PK_j, g)}{e(R, g)} \cdot e(g, P_U)^{\bar{k}_1}\right)^{H_0(PK_j)} \\ & \stackrel{?}{=} e(P_{pub}, g)^{s_2 H_0(PK_j) - s_3} \end{aligned} \quad (1)$$

4) If Equation (1) holds, it breaks it.

}

If $j \leq n$ then it outputs the identity index j of the real signer. Otherwise, it outputs False.

We will show that our attack is valid since if the real signer's public key is PK_s, we can obtain the following relation

$$\begin{aligned} k_1 &= s_1 - c \cdot H_0(PK_s) \quad (2) \\ s_2 H_0(PK_s) - s_3 &= k_2 H_0(PK_s) + cr H_0(PK_s) \\ &\quad -cr H_0(PK_s) + x_s \\ &= k_2 H_0(PK_s) + x_s \quad (3) \end{aligned}$$

Thus, we have

$$\begin{aligned} e(P_{pub}, g)^{s_2 H_0(PK_s) - s_3} &= e(P_{pub}, g)^{k_2 H_0(PK_s) + x_s} \\ &\Updownarrow \\ e(P_{pub}, g)^{s_2 H_0(PK_s) - s_3} &= e(P_{pub}, g)^{k_2 H_0(PK_s)} e(P_{pub}, PK_s) \\ &\Updownarrow \\ e(P_{pub}, g)^{s_2 H_0(PK_s) - s_3} &= \left(\frac{\Pi \cdot e(g, P_U)^{k_1}}{e(R', g)}\right)^{H_0(PK_s)} \\ &\quad \cdot e(P_{pub}, PK_s) \\ &\Updownarrow \\ e(P_{pub}, g)^{s_2 H_0(PK_s) - s_3} &= \left(\frac{\Pi \cdot e(g, P_U^{k_1} \cdot PK_s)}{e(R, g)}\right)^{H_0(PK_s)} \\ &\quad e(P_{pub}, PK_s). \end{aligned}$$

It means that the real signer's public key PK must satisfy Equation (1). Thus our attack is valid.

The reason to produce such attack is that random number k_1 in signing phase can be recovered by making use of the hash value of $m||V||P_{pub}||\Pi||R$ and the actual signer's identity PK_s. At the same time, s_2 and s_3 in the ring signature exist a certain relevance. It makes that the relation $s_2 H_0(PK_s) - s_3 = k_2 H_0(PK_s) + x_s$ holds. Thus, it reveals the relevant identity information of the actual signer.

4.2 Attack on Unforgeability

For a ring signature, unforgeability should be a very important property, namely, it is difficulty to forge a ring signature on a new message m^*. The property ensures that ring signature can provide stronger unforgeability. Unfortunately, by analyzing the security of their scheme, we show that Qin *et al.*'s ring signature scheme also does not satisfy unforgeability. The detail attack is given as below.

1) Let m^* be a forged message. $L = \{PK_1, \cdots, PK_n\}$ is a public key list.

2) First, the attacker picks a random number $\alpha \in Z_q$ to calculate $P_U^* = g^\alpha$.

3) Then, it chooses three random numbers $r_1, r_2, r_3 \in Z_q^3$ to calculate

$$\Pi^* = e(g, P_U^*)^{-r_1} e(P_{pub}, g)^{r_2} e(g, g)^{r_3} e(R, g)$$

where $e(R, g) = \prod_{j=1}^{n} e(PK_j, g)$.

4) Next , it randomly selects $v \in Z_q$ to set $V^* = v$, and computes $c^* = H_1(m^*||V^*||P_{pub}||P_U^*||\Pi^*||R)$, where $R = \prod_{j=1}^{n} PK_j$.

5) Subsequently, the attacker sets $s_1^* = r_1$, $s_2^* = r_2 + \alpha \cdot c^*$ and $s_3^* = r_3 - V^* \cdot c^*$.

6) Finally, the resultant ring signature on message m^* is $\delta^* = (c^*, P_{pub}, P_U^*, V^*, s_1^*, s_2^*, s_3^*)$.

In the following, we show that the above forged signature δ^* is valid, that is to say, it can pass the verification checking. Because

$$\begin{aligned} \Pi' &= e(g, P_U^*)^{-s_1^*} e(P_{pub}, g)^{s_2^*} e(g, g)^{s_3^*} e(P_{pub}, P_U^*)^{-c^*} \\ &\quad e(g, g^{V^*})^{c^*} e(R, g) \\ &= e(g, g^\alpha)^{-r_1} e(P_{pub}, g)^{r_2 + \alpha c^*} e(g, g)^{r_3 - V^* c^*} \\ &\quad e(P_{pub}, g^\alpha)^{-c^*} e(g, g^{V^*})^{c^*} e(R, g) \\ &= e(g, g^\alpha)^{-r_1} e(P_{pub}, g)^{r_2} e(g, g)^{r_3} e(R, g) \\ c^* &= H_1(m^*||V^*||P_{pub}||P_U^*||\Pi^*||R). \end{aligned}$$

It means that our forged ring signature is valid since it can pass the verification equation. Thus, Qin *et al.*'s scheme is insecure. Any one can forge a ring signature on arbitrary a mesasage. Thus, our attack is valid.

In the following, we analyze the reason to lead to forgery attack. For the verification algorithm of the ring signature, we can find that any one can calculate Π' by random choosing $(s_1, s_2, s_3, c, P_U, V)$. Essentially, Π' is irrelevant to the hash value c since Π' can be computed without c by choosing the appropriate (s_1, s_2, P_U, V, s_3). Thus, the main reason to produce such attack is that the generation of Π is independent of hash value c. c can not restrain the generation of Π.

5 Conclusions

In this letter, we analyze the security of *Qin et al.*'s PCRS scheme, and show that their scheme is insecure. It can not achieve two security properties of ring signature: **unforgeability** and **anonymity**. In their scheme, any one can forge a ring signature on arbitrary a message, and given a ring signature δ, the verifier can know who is the actual signer. Finally, our analysis is confirmed thought two concrete attacks, the corresponding reasons to produce such attacks are given. It is our future work how to design a secure and practical ring signature scheme with constant-size.

Acknowledgments

This research was was supported by Beijing Municipal Natural Science Foundation (Nos.4162020), Guangxi Key Laboratory of Cryptography and Information Security (No.GCIS201710) and Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (No.MIMS16-01).

References

[1] M. Bellare and G. Neven, "Multi-signatures in the plain publickey model and a general forking lemma," in *Proceedings of the 13th ACM Conference Computer and Communications Security*, pp. 390-399, 2006.

[2] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *Proceedings of IEEE Symposium on Security and Privacy (SP'14)*, pp. 459–474, May 2014.

[3] D. Bleichenbacher and U. Maurer, "On the efficiency of one-time digital signatures," in *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'96)*, pp. 176–180, June 2008.

[4] P. Bose and C. P. Rangan and D. Das, "Constant size ring signature without random oracle," in *Proceedings of the 20th Australasian Conference Information Security and Privacy*, pp. 230–247, July 2015.

[5] N. Chandran, A. Sahai and J. Groth, "Ring signatures of sublinear size without random oracles," in *International Colloquium on Automata, Languages, and Programming (ICALP'07)*, pp. 423–434, vol. 4596, June 2007.

[6] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup, "Anonymous identification in ad hoc groups," in *Proceedings International Conference Theory and Application Cryptographic Techniques*, pp. 609–626, 2004.

[7] R. Ehmet, L. Z. Deng, Y. Y. Zhang, and J. W. Zeng, "Multi-proxy multi-signature without pairing from certificateless cryptography," *International Journal of Network Security*, vol. 20, no. 3, pp. 403–413, 2018.

[8] E. M. Ghadafi, "Sub-linear blind ring signatures without random oracles," in *Proceedings of the 14th IMA International Conference Cryptography and Coding (IMACC'13)*, pp. 304–323, Dec. 2013.

[9] S. Goldwasser, C. Rackoff and S. Micali, "The knowledge complexity of interactive proof systems," in *SIAM Journal on Computing*, pp. 186–208, 1989.

[10] M. S. Hwang and I. C. Lin, *Introduction to Information and Network Security (6ed, in Chinese)*, Taiwan: McGraw Hill, 2017.

[11] M. S. Hwang and C. Y. Liu, "Authenticated encryption schemes: Current status and key issues," *International Journal of Network Security*, vol. 1, no. 2, pp. 61–73, 2005.

[12] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves", *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 73–84, 2004.

[13] L. Nguyen, "Accumulators from bilinear pairings and applications," in *The Cryptographers Track at the RSA Conference (CT-RSA'05)*, pp. 275–292, 2005.

[14] R. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565, vol. 2248, June 2001.

[15] J. H. Zhang and Q. C. Dong, "Efficient id-based public auditing for the outsourced data in cloud storage," *Information Sciences*, vol. 343-344, no. 2, pp. 1–14, 2016.

[16] J. H. Zhang, P. Y. Li, and M. Xu, "On the security of a mutual verifiable provable data auditing in public cloud storage," *International Journal of Network Security*, vol. 19, no. 4, pp. 605–612, 2017.

[17] Y. L. Zhao, M. J. Qin and Z. J. Ma, "Practical constant-size ring signature," *Journal of Computer Science and Technology*, vol. 33, no. 3, pp. 533-541, 2018.

Biography

Jianhong Zhang received his Ph.D. degrees in Cryptography from Xidian University, Xian, Shanxi, in 2004 and his M.S. degree in Computer Software from Guizhou University, Guiyang, Guizhou, in 2001. He was engaging in postdoctoral research at Peking University from October 2005 to December 2007. He has been an Assistant Processor of College of Sciences, North China University of Technology, Beijing China, since 2001. His research interests include computer networks, cryptography, electronic commerce security, computer software.

Wenle Baig received his Ph.D. degrees in Communication Networking from Beijing University of Posts and Telecommunications; , Beijing, in 2006 and his M.S. degree in Computer Software from Beijing University of Posts and Telecommunications,Beijing in 2001. He has been an Assistant Processor of College of Sciences, North China University of Technology, Beijing China, since 2001. His research interests include computer networks, cryptography, electronic commerce security, computer software.

Zhengtao Jiang received his Ph.D. degrees in Cryptography from Xidian University, Xian, Shanxi, in 2005 and his M.S. degree in School of Mathematics from Central South University, Changsha, Hunan, in 2002. He was engaging in postdoctoral research at Beihang University from October 2008 to December 20010. He has been an Assistant Processor of School of Computer Science, China Communication University, Beijing China, since 2001. His research interests include computer networks, cryptography, electronic commerce security, computer software.

Timestamp Based Detection of Sybil Attack in VANET

Syed Mohd Faisal and Taskeen Zaidi
(Corresponding author: Taskeen Zaidi)

Department of Computer Science and Engineering, Shri Ramswaroop Memorial University
Village Hadauri, Post Tindola, Lucknow - Deva Road, Barabanki, Uttar Pradesh 225003, India
(Email: taskeenzaidi867@gmail.com)

(Received July 24, 2019; Revised and Accepted Dec. 15, 2019; First Online Feb. 28, 2020)

Abstract

VANET is a subset of MANET in which communication among the vehicles may be done using vehicle -to- vehicle or roadside infrastructure. But there may be chances of attacks in VANET due to mobility of nodes and random change in topology. One of the prominent attack is Sybil attack in which attacker creates multiple false identities to disturb the functionality of VANET. In literature many solution have been proposed for detection and protection of vehicles from Sybil attack. In the current work, authors have proposed a Sybil node detection technique based on timestamp mechanism. In this work timestamp is a unique certificate provided by RSU to all vehicles on the road in VANET. In the proposed work for node discovery and data transmission, we used Ad-hoc On Demand Distance Vector (AODV) Routing protocol and timestamp as a hash function of public key and for detection of the Sybil node implemented through NS2 simulator.

Keywords: Network Security; NS2; NS3; Sybil; VANET

1 Introduction

With the recent development in network topologies and trends, VANET receives a lot of interest of researcher and scientists.

Vehicular Ad-hoc Network (VANET) is a shade of Mobile Adhoc Network, which has potential to improve passenger safety by means of communication among vehicles [10]. Mobile Ad-hoc Network (MANET) provides communication between stand by devices, these devices either have slow movement or no movement on the contrary VANET establish and cater communication betwixt swift moving vehicles [20]. Apart from this, their are several other differences between MANET and VANET which were listed in Table 1.

Each year exponential number of vehicles were running on roads, in course of time, definitely, it will grow drastically. It will cause traffic on the roads that fritter away time, money, petroleum products and many more. Government is investing more money to construct more and more roads and destroying landscape for the reason that existing roads are not in a condition to support generated traffic. Their is a solution of all these problems *i.e.* VANET. VANET provides communication between vehicles that helps to improve traffic on the roads. Prior to, vehicles obtain traffic information of route and driver takes decision based on that information.

VANET is a way to enforce Intelligent Transport System (ITS) based on IEEE 802.11p standard for the Wireless Access for Vehicle Environment (WAVE).

VANET allows vehicles to seamlessly connect in a region, where no existing infrastructure is available. VANET system aims at providing platform for various services that can improve passenger's safety and efficiency, driver assistance, infotainment, transportation regulation *etc.* In order to provide these services VANET require accurate and timely (no delay) data transmission. Information is transmitted in between Vehicles (mobile nodes) and some nearby stable Road Side Units (RSU).

VANET connects those vehicles that were in range of 100 to 500 meters and if a vehicle is not in that range, it will not connect with other vehicles. Figure 1 shows the complete architecture of VANET.

Primarily VANET supports two types of communication *i.e.*

1) Vehicle-to-Vehicle (V2V) Communication;
2) Vehicle-to-Infrastructure (V2I) Communication.

Vehicle-to-Vehicle (V2V) communication do not require any infrastructure for communication, it creates a self-network. The range of this network is up to 500 meters from the geographical position of initiator vehicle. Vehicle in that range can communicate using On Board Unit (OBU) with other vehicles of the network sans availing the benefit of VANET. In V2V, vehicle share information about safety messages, vehicle identities and information about malicious vehicle. On contrary, Vehicle-to-Infrastructure (V2I) communication relies totally on the preset infrastructure *i.e.* Road Side Unit (RSU). In

Table 1: Comparison of MANET with VANET

SN	Parameters	MANET	VANET
1.	Node Mobility	Low	High
2.	Node Density	Low	High
3.	Change in Network Topology	Slow	Frequently
4.	Energy Constraints	Medium	Low
5.	Moving pattern of Nodes	Random	Constrained by Road
6.	Range	100m	Upto 500m
7.	Node Speed	Low(6km/hr)	Medium-High(20-100Km/hr)
8.	Scalability	Average	High
9.	Bandwidth	Hundred Kbps	Thousand Kbps
10.	QoS	Low	High

V2I communication RSU communicate with Vehicles and transmit network management message, road condition, nearby hotel, internet access. Figure 1: Architecture of VANET shows the broader picture of VANET and services provided by it.

As we know that movement of vehicles are very fast in VANET so the intensity of connection and disconnection of vehicles are very high and due to that topologies keeps changing. Meanwhile, there is a favorable chance for attacker to attack on the network when the topologies change. Therefore, security is also a major concern in VANET, it is necessary to figure out the chances of attack and eliminate them.

Firstly, VANET is a wireless network so it inherits all the security threats of wireless network. Secondly, movement of vehicles are high in VANET so for the efficient communication vehicles keeps switching between topologies because of that there is a great chance of attack at the time of handoff.

Various forms of forging attacks were implemented and designed by authors [7]. The attacks were analysed through vehicles speed, number of collisions and percentage of delivered packets. Rana *et al.* [21]. modified PKI for message authentication, integrity and privacy. A coordination based algorithm designed for dynamic network for information flow and vehicle security is maintained by a unique signature method [2]. Grover *et al.* [6] used neighborhood list to detect and identify Sybil node. This scheme specifies that if neighboring node is watching a malicious node for a longer duration of time then this node will be identified as a Sybil node but the drawback of this scheme was that it was very complex and time consuming. A survey routing protocols for VANET has been done by authors [9] on unicast, multicast, geocast, mobicast and broadcast protocols. A navigation based system for VANET is proposed for the guidance of drivers in real-time manner. The system is useful for computation of better route in real road based scenario [1]. A system model was proposed by authors [25] using dynamic certificate generation technique to restrict and identify the Sybil node. It was analyzed by authors [14] that in VANET vehicles communicated through RSU using central server and road side servers. The communication may be Vehicle to Vehicle (V2V) or Vehicle to Roadside (V2R).

Authors [23] proposed a shared key management technique which has advantage over distributed key management system. In this framework the vehicles may be inter connected automatically without using RSU. The message transmission is done by RSU. Emergency Electronic Brake Lights System was proposed by authors [3] which warns the vehicles on the road about weather condition and V2V communication is used to propagate the alert messages to the vehicles. A distributed and localized approach was proposed by authors [30] for the detection of Sybil nodes on roads. Two algorithms were proposed for position verification and detection of Sybil attacks by observing signal strength. Simulation was performed to analyze the proposed scheme. Authors [8] have proposed a Sybil node detection method using electro acoustic position using context aware switching technique. Simulation has been done to analyze the accuracy of proposed scheme. A physical layer authentication scheme was proposed by authors [29] to detect Sybil attack in indoor and urban environment. Hypothesis was proposed to detect the Sybil nodes for narrowband and broadband wireless system. The performance was depicted through network analyzer tool. NS2 is a networking tool used for simulation purpose for wired and wireless networks [13, 18].It is an open source simulator used for networking research and has support protocols like TCP, FTP, UDP, HTTPS, DSR and AODV. It uses TCL as an scripting language and C++ and OTCL language.

2 Background

Because of the communication in wireless environment, VANET is exposed to various attacks and threats as shown in Figure 2. Following are the necessities required to ensure security in VANET [11, 15].

Authentication.

Authentication framework is vital to identify as it ensures that the participants in the network is as same

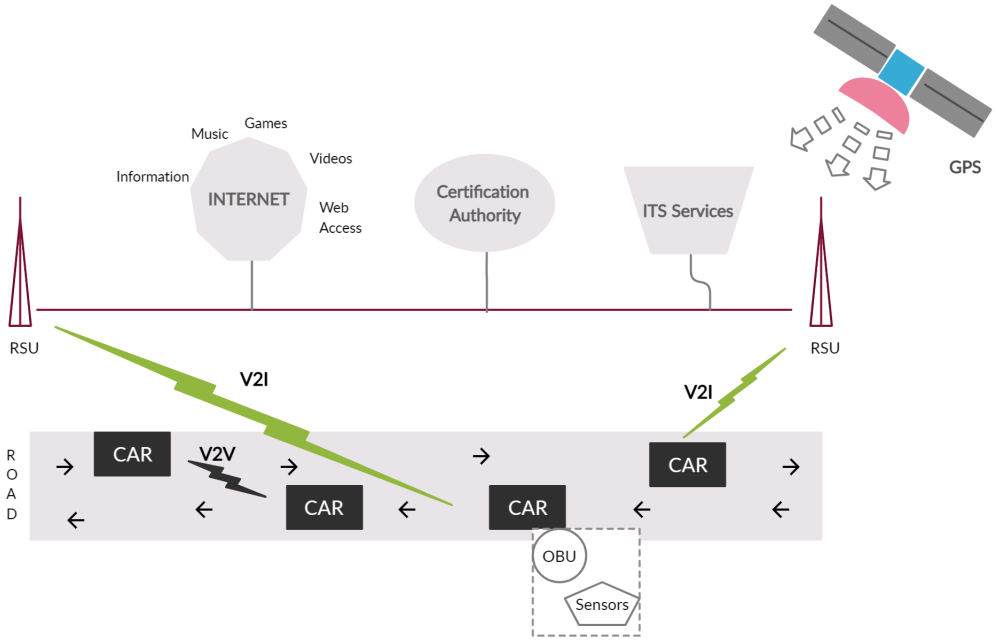


Figure 1: Architecture of VANET

as it claim to be. It also certifies that authenticated vehicle gain all the privilege provided by VANET.

Integrity.
Integrity of the messages should be preserved *i.e.* transmitted messages are prohibited to alter in the transmission medium [19].

Availablity.
Network resources should be available even in the time of failure and in the presence of malicious node. Availability of the network is directly related to all the other security attributes. Even in the worst case, network should be available and run efficiently.

Confidentiality.
Confidentiality is not to share private information with adversaries. Not every message in the network needs to preserve confidentiality but messages containing information like session key, payment data, OTP needs to be secure and guard confidentiality. Requirement of confidentiality is needed when transmitting some data or when multiple node comes in-group communication. In both the cases if confidentiality breaks then malicious vehicle may take maximum privilege of the network or may damage the network integrity and availability.

Non-Repudiation.
Non-Repudiation assures that someone cannot deny the validity of something. Typically nonrepudiation refers that sender of the message cannot deny the authenticity of their signature on a message. It is important to resolve dispute about who transmitted this message.

Privacy.
Most of the information about the nodes are broadcasted publicly in VANET so it is necessary to maintain privacy. Communication in the network should be anonymous and message sent by authorized vehicle should be protected in the presence of unauthorized observer *i.e.* authenticated nodes/vehicles have right to access personal information [17]. Contrarily adversaries may collect and analyze information, fix up a trap and harm the user.

Scalability.
Scalability refers to the capacity of a network to manage the growing vehicles and network. Scalability often refers to the stability of the network so that network performance will be not disrupted or degraded even in the worst case.

3 Classification of Attackers

In this section, we classified the attackers based on its behaviour, nature and efficiency. Efficiency of all attacks depends on the capacity of attackers [16, 22, 28]. Therefore, before discussing attacks it is essential to know about the type of attackers as shown in Figure 3.

Active vs. Passive.
Some attackers do not transmit or receive any message on the network, though they eavesdrop on the wireless network to gain knowledge about the pattern and frequency of the data transmission and use this earful information in future. These attacks are done by Passive Attackers, on the contrary, Active

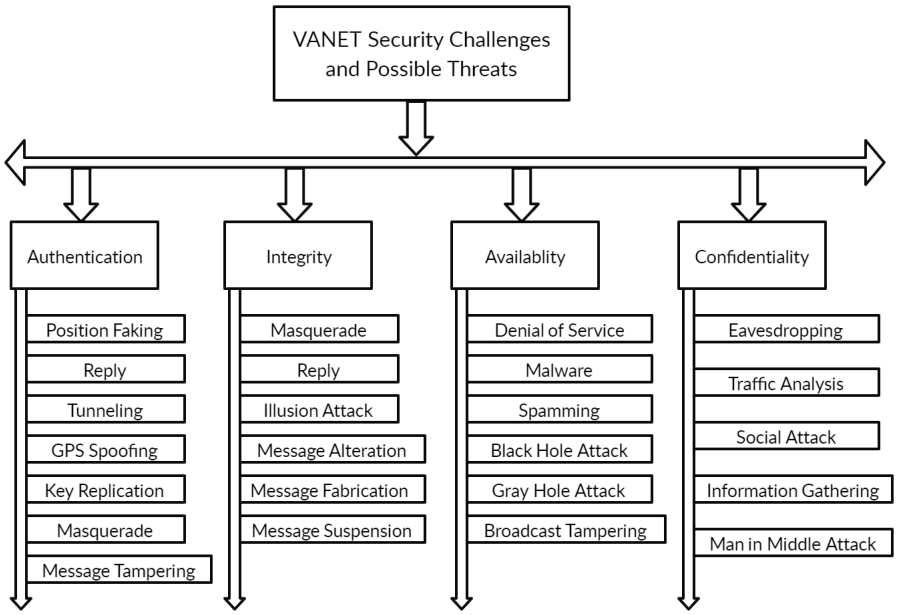


Figure 2: VANET Security Challenges and possible attacks

Attackers alter the information it receives, generate false signals, do not forward received packets, apply modification in data stream to disrupt the efficiency of the network or to gain access of unauthorized services.

Insider vs. Outsider.
An attacker may be an authorized member of the network who has all the knowledge and access of the network, such attackers are known as Insider. While Outside Attackers (outsiders) are intruders, who do not have access to communicate directly to the insiders and can launch attacks of less variegation.

Malicious vs. Rational.
Not all attacks are launched to seek personal benefits; some attacks are launched to disrupt the performance of network and to create hurdle for the members of the network, these attackers are known as Malicious Attackers. On the other hand, Rational Attackers seeks personal benefits; attackers launch these attacks intentionally for specific node or for specific network.

Local vs Extended.
Local Attackers launch attack of limited scope and in limited control region/area. Whereas, attackers of extended class controls several entities, which are distributed across whole network. Extended class attackers have potential to degrade the performance of the network or shut down the entire network.

4 Sybil Attack

VANET works on wireless environment, due to which it is vulnerable to many types of the security attacks. Because of the unique nature of VANET, it adds additional vulnerability and complexity in order to create a secure network. There were many threats possible on VANET but in this paper, we will focus on Sybil Attack as it is the root cause of many attacks and security threats. Sybil attack was first introduced and illustrated by Douceur in context to Peer-to-Peer Network [7, 26].

Sybil attack is a threat against security of a network. As malicious vehicle, impersonate multiple legitimate identities by forging new identity or by stealing identities from vehicles of the network. Attackers steal identities of other vehicles by eavesdrop broadcasting messages, as the vehicle are highly mobile in nature, density of network changes dynamically, network topology also changes dynamically so vehicles continue to communicate with other in order to update their routing tables. Attackers take the advantage of these properties and impose Sybil attack over the vehicles of the network and create the illusion of presence of multiple legitimate vehicles in the network. In the Figure 4, victim vehicle (green node) is surrounded by multiple Sybil vehicle (black nodes) that can block all the transactions by performing attack it receives access over the vehicle.

Sybil vehicle have potential to influence the functioning of the services of VANET like update routing table, voting, fair resource allocation, misbehavior detection, data aggregation, *etc.* By imposing Sybil attack, attacker takes over the control of network and may inflict other attacks such as black hole attack, timing attack, denial of service attack, impersonation attack and others.

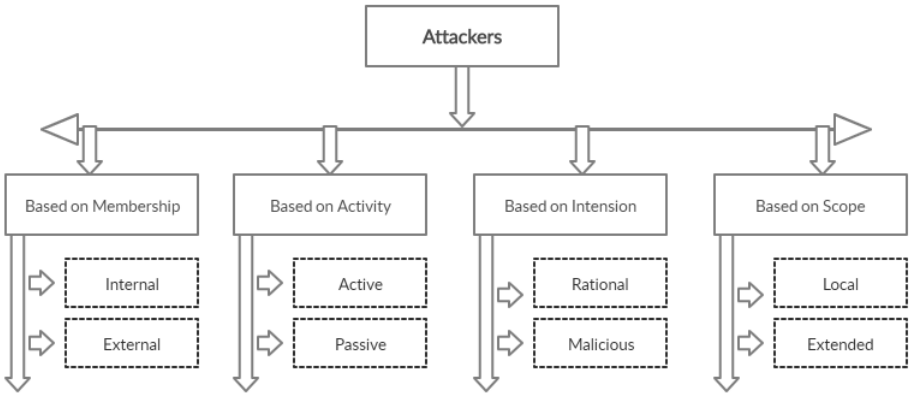


Figure 3: Attackers classification

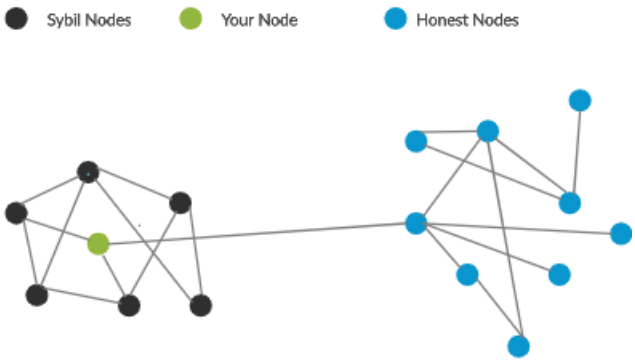


Figure 4: Sybil attack

For Example: Suppose a vehicle on highway significantly reduced its speed and broadcast a warning message. Recipients of the message will reply it and further broadcast that message. However if there are Sybil vehicles, then those Sybil vehicles can deny to reply and broadcast (DOS attack). This can invert the proper functioning of the network and create a massive pileup on the highway and serge loss of life. In addition, Sybil vehicle in network can impose any attack on the vehicle or network.

5 Analysis of Defence Mechanism

We know that there is no logical central authority for the efficient functioning of VANET, many protocols award unique identity to vehicles and then apply security rules and measures to defend from Sybil attack [5]. Although researchers proposed different mechanism to secure network from Sybil Attack:

1) Resource testing method;
2) Position verification method;
3) Domain specific;
4) Trusted devices;
5) Trusted certification method;
6) Neighbor list method.

5.1 Resource Testing Method

This method is used to detect Sybil attack. Resource testing method test vehicle resources, *i.e.* radio resource, memory resource, identification resource and computational resource. It is assumed that every vehicle is equipped with limited computational resources. In this method, a typical puzzle is distributed to the vehicle of the network in order to check the computational ability of vehicles. A Sybil attacker vehicle also receives the puzzle for computation but because of handling multiple identities with limited computational resources, it is impossible for Sybil vehicle to perform additional computation. This approach is based on assumption that every vehicle has same and limited computational resources but there is a chance that Sybil vehicle may have additional computational resources. Therefore, this method is not suitable for the detection of Sybil attacker.

5.2 Position Verification Method

Sybil attack detection using position verification method is based on the fact that a vehicle can present only on one position at a time. In this method physical location of any vehicle is to be verified before the data transmission. Network transmits a request message to all the vehicles of the network. All vehicles on the network are bound to respond that message; here Sybil vehicle also transmits their position coordinates. As network receives exactly same geographical coordinates from multiple vehicles which reflects that a particular vehicle is behaving like a Sybil vehicle and after the detection of Sybil vehicles network takes necessary action for the elimination of Sybil vehicle from the network.

5.3 Domain Specific

Some researches proposed Sybil attack detection in light of domain. Proposed Sybil attack detection mechanism is based on the location/geographical position of vehicles. If an attacker vehicle having single device and attacker starts performing Sybil attack then all the Sybil vehicles move together in specifically same fashion with same speed in a specific domain. In this way, network track down the trajectory and pattern of the Sybil vehicles and generate alert signal. However, this method is sufficient to track the Sybil vehicles having unary device but this method is not efficient against malicious vehicles having multiple devices.

5.4 Trusted Devices

In this method, trusted devices are combined with trusted Certificates, the binding of hardware with vehicles restrict vehicles to obtain multiple false keys. This method is sufficient to secure the network from Sybil attack, but it pushed an overhead of installation of extra hardware in vehicles, another issue is that there is no such efficient mechanism that restricts vehicle from obtaining multiple trusted devices except the manual interaction.

5.5 Trusted Certification

Trusted Certification is the most commonly used solution for the prevention of Sybil attack due to its ease of implementation and potential to remove Sybil attack from the network. In this method, a third party certification authority is responsible for issuing identities and Centralized authority assigns these identities to vehicles. Centralized authority also ensures the uniqueness of certificate for every vehicle. There is no such mechanism for issuing unique identities to every vehicle; it is to be done manually. There is no big deal in issuing certificate but it creates a bottleneck in large scale system, another issue is to manage a database for used, unused, lost and stolen identities. Because of these issues, it is difficult to implement this mechanism although this mechanism is efficient and removes the headache of installation of new hardware in vehicles.

5.6 Neighbor List Method

Author proposes a new method for the detection of Sybil vehicle using list of neighbor vehicle. This approach is based on assumption that if a vehicle is observing same neighbor vehicles simultaneously for long duration of time, that means there must be a Sybil vehicle. In order to obtain information about neighbor vehicles, every vehicle keeps exchanging their information with neighbors and vehicle create list of neighbor vehicles. This scheme is complex as after every time interval (T1, T2, T3..) vehicles share list of neighbor vehicle to detect Sybil vehicle. Here intersection operation is performed on the list of neighbor vehicles generated at different intervals and mark suspected vehicles *i.e.* particular vehicle is neighbor vehicle for seamlessly long duration of time. This approach put an extra overload by sharing list of neighbor vehicles but sill fails to detect Sybil vehicle in some cases.

Case 1: Suppose a vehicle in a network refuses to share a list of neighbor vehicle.

Case 2: Sybil vehicle claims itself as not a neighbor vehicle of any vehicle. In both the cases, intersection operation failed to detect suspected vehicles, as there is no mechanism that bound vehicles not to perform these malicious activities [4, 27].

6 Comparative Study

6.1 Resource Testing Mechanism

Resource Testing Mechanismwas not sufficient to detect and prevent Sybil attack, as it is restricted to identify fake identities and in some case it fails when attacker will equip its vehicle with extra computational equipment.

6.2 Position Verification

Position Verification mechanism requires extra hardware instead, there is no such guarantee that all the vehicles are legitimate. Any attacker vehicle may change its geographical coordinates and forward that information, in that case there is no such mechanism to identify and track down the actual geographical coordinates of vehicles.

6.3 Domain Specific

Domain Specific is a better technique to detect attacker vehicles although it gets restricted and generates falsie result in case when vehicle creates its replica.

6.4 Trusted Devices

Trusted Devices requires installation of extra hardware that increases the cost of vehicles but there is no central authority, which makes sure that every vehicle is equipped with only one trusted device. By the reason there are several chances of a vehicle, equipped with multiple trusted devices and in that scenario Trusted Device mechanism fails to detect and eliminate Sybil vehicles.

6.5 Trusted Certification

Trusted Certification mechanism requires a huge amount of data transmission in order to validate a vehicle. This mechanism blindly relies on the third party who is issuing certificate to vehicles moreover if Sybil attacker directly attacks over the certification authority then the whole system shuts down, consequently attacker can gain full privilege of the VANET. Another issue is this, that there

is no proper mechanism to identify lost and stolen certificates, because of this, identification of false certificate requires a good amount of computation power and time.

6.6 Neighbor List Method

Neighbor List Method was not sufficient when Sybil vehicles refused to participate or transmit false data to the leader vehicle. Another issue was that Neighbor list method requires a lot of computation power and by the time there are great chances for an attacker to commit an attack, disrupt the whole system and leave the network.

7 The Proposed Schema

In this section, we proposed a method to detect Sybil attack. Proposed schema used timestamp mechanism for the detection of Sybil vehicle in VANET. This mechanism is ideal for less number or average number of vehicles. As we know there were number of limitations in VANET *i.e.* confidentiality, integrity, repudiation *etc.* Therefore, the proposed mechanism is designed keeping all these issues in knowledge.

Way for the detection and elimination of Sybil vehicle, we used timestamp mechanism. Time Stamp is a unique certificate, which is provided by Road Side Unit (RSU) to all the vehicles on the road that want to take privilege of VANET. Time stamp is a unique identity but here we assume time stamp as a Hash Function of Public key and for the sake of security, only RSU know the Hash key and has authority to generate, assign time stamp to requesting vehicles.

It is to be assume that

1) Certification Authority (CA) assigns some unique public key to manufacturers and manufacturers assign these keys to vehicles, which is hard-coded into vehicles communication device. In this way, every vehicle receives a registered unique public key.

2) It is impossible for a vehicle to pass from multiple RSU at same time.

Taking these assumptions in knowledge, we start our workflow for the detection of Sybil attack and later we will provide pseudo code of our proposed mechanism.

Vehicular Ad-hoc Network is a secure network *i.e.* without complete authentication and verification, vehicles are not allowed to access VANET services. For the initial authentication purpose, vehicles transmit its public key (Pki) to RSU, as vehicles enter into VANET environment.

RSU authenticate public key from Certification Authority (CA): As Certification Authority has a list of all keys issued to vehicles. If Certification Authority sends an acknowledgement (ACK) to RSU then RSU will generate Time stamp (hash function of requesting vehicle public key) and issue a Time stamp to vehicle. On the flip side if Certification Authority issue Negative Acknowledgement (NAK) to RSU, then RSU wont issue Time Stamp and block the request for specific period of time, still if RSU continuously receives request message from same public key then RSU mark requesting public key as suspected key, generate an alert message and broadcast that public key as suspected key.

As soon as vehicle receives Time Stamp issued by RSU, vehicles get license to access VANET services and communicate with other vehicles. For the communication VANET vehicles are bound to insert its own Time stamp into message packet and update hop count.

Table 2: List of Timestamp assigned

S.No	Public Key	Timestamp (hash function of Public key)
1	ck1213n	dkjj1152cmwchb
2	zzaj132	bbdcdc55155njd
3	.....	.....
.	.....	.....

When any vehicle receives the requesting packet, it finds out for the destination into its routing table and responds if destination is found otherwise intermediate vehicle/this vehicle decreases the hop count, update its timestamp, public key into message packet and flood the packet into network. As soon as destination is found, destination vehicle inserts its own time stamp into message packet and revert the message to the source vehicle. By the time, search packet is on the network source vehicle wait for respond only for time interval t (varies with protocol) if source vehicle will not receive the respond within time, it will rebroadcast the request message. On the other hand if source vehicle receives the respond (that enclose vehicle id and time stamp of all intermediate vehicles) it will forward vehicle id and time stamp of destination vehicle to RSU. As we saw earlier only RSU keeps mapping public key of requesting vehicle so it maintains the table of time stamp assigned to corresponding public key. As soon as RSU receives Public Key and Time stamp from source vehicle it will look for its table of assigned time stamp to respective public key as shown in Table 2. If timestamp is same as assigned by RSU to corresponding public key, RSU forward acknowledgement packet to source vehicle and source vehicle start transmitting message via same intermediate vehicles. If RSU send negative acknowledgement then source vehicle discard the message packet and again flood fresh request query for the destination.

When destination receives the message from the source vehicle, it will not directly accept that message packets although destination vehicle forward the list of source vehicle and intermediate vehicles public keys with their corresponding time stamp to RSU, if all the time stamps are

valid as assigned by RSU then RSU forward the acknowledgement to destination consequently destination vehicle receives/process all the messages.

Assuming that while matching, RSU founds any timestamp which would not assigned by any RSU to specific public key or RSU founds similar public key with multiple timestamps or vice versa in all these cases RSU forwards negative acknowledgement to Destination vehicle, mark that public key as Sybil Vehicle, generate alert and forward the updated list of Sybil Vehicles to neighbor RSU. Destination vehicle discard all the messages and forward the request for the retransmission of messages.

By using this mechanism we validate all the vehicles involve in communication *i.e.* source, intermediate and destination vehicles and eradicate all the malicious vehicles meanwhile we authenticate the integrity and confidentiality of the messages send and received. At end we identified the Sybil vehicles and block the vehicles for the future perspective.

Suppose a vehicle ($V_i = S_0$) wants to communicate with other vehicle ($V_j = DS_1$), then source vehicle (V_i) search for destination in its routing table. For an instance, destination (DS_1) founds then source vehicle (V_i) authenticate the validity of Destination vehicle through RSU if RSU forward Acknowledgement (ACK) then Source vehicle encrypt the request message and transmit the message to Destination vehicle (DS_1). If destination not found, then source vehicle (V_i) appends its public key (Pk_i), Time stamp (T_i), Hop Count (HC), Destination address in request message and rebroadcast the request message to neighbor vehicles. Neighbor vehicle checks its routing table, if there is an entry of destination vehicle (DS_1) or neighbor vehicle itself is a destination vehicle then in both the case respective vehicle replies to Source Vehicle (V_i) about the path otherwise neighbor vehicle append its public key (Pk_k), Time stamp (T_k), decrease the hop count value and broadcast again to its neighbor vehicle. This process continues until destination is not found or value of hop count becomes zero.

As destination vehicle (DS_1) receives the request message, decrypt the packet using its private key and start the request packet authentication process. As we know every request packet contains public key and timestamp of all intermediate vehicles, so destination vehicle transfers list of public keys and time stamps to RSU. As RSU issued time stamp to vehicles, so it maintains a data base of time stamp issued to public key of vehicles. RSU compares the list of time stamp and public key forwarded by destination vehicle with its own time stamp issued to respective public keys. If RSU found all legal timestamp issued to public key, it replies Acknowledgement (ACK) to destination vehicle (DS_1). Moreover, in case RSU founds some illegal/wrong time stamp over corresponding public key, RSU replies Negative Acknowledgement (NAK) to destination (DS_1) and mark the public key as Sybil vehicle, add the malicious public key in list of suspected vehicle, generate alert and transmit the updated list of suspected vehicle to all their neighbors. If Destination vehicle receives NAK (form RSU), then it discards the request message and request for the retransmission of that message.

Only when destination vehicle (DS_1) receives the Acknowledgement (ACK) message form RSU, then destination receives the message and process that message sent by source vehicle.

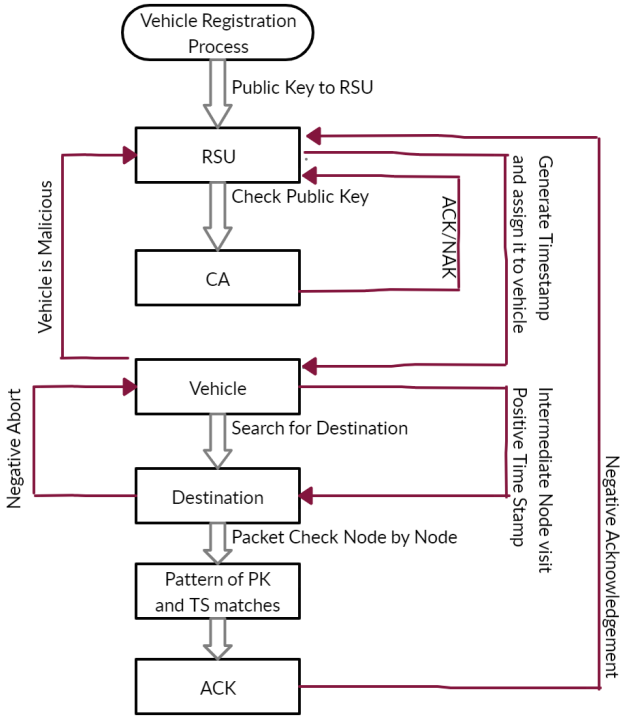


Figure 5: Flowchart of Proposed algorithm

7.1 Update Time Stamp

For the communication in VANET, vehicles are bound to have time stamp so as vehicles enters into VANET environment RSU issues time stamp over its public key and RSU update the database of time stamp issued. Usually vehicles are in mobile fashion and passes from multiple RSUs but it not possible to pass from multiple RSUs at a time interval. When a new vehicle crosses a RSU, vehicle forward its public key to RSU, RSU authenticate the public key from Certification Authority (CA), if RSU receives ACK from CA then RSU generates hash function from public key and forward the hash function as assigned time stamp to the requesting vehicle. Otherwise, RSU generate an alert and mark the public key as suspected vehicle. Suppose that a previously registered vehicle crosses new-RSU in such case vehicle forward time stamp and public key to new-RSU, as we know neighbor RSUs have list of timestamp issued over public key. Thus new-RSU authenticate the time stamp issued by previous-RSU and this authentication will take place via neighbor

RSUs, if authentication is successful then new-RSU issues new time stamp and assigns new time stamp to requesting vehicle, in the mean while new-RSU update the list time stamp issued and forward the list of updated time stamp to neighbor-RSUs. In the same way, every vehicle gets authenticated and receive updated time stamp. This process is also helpful in rectifying Sybil vehicle as this process issues only one time stamp to authenticated public key so when Sybil vehicle insult false public key, RSU generate an alert and mark the public key/ vehicle as suspected public key/vehicle.

Figure 5 shows the flowchart of proposed algorithm for the detection and elimination of Sybil vehicle.

7.2 Algorithm

Table 3 shows the pseudonym of notations used in Algorithm 1.

Table 3: Notations used in this paper

S.No	Notation	Comments
1.	V_i	ith vehicle
2.	RSU_j	jth RSU
3.	ACK	Acknowledgement message
4.	NAK	Negative Acknowledgment
5.	Pk_i	Public Key of Vehicle i
6.	S_0	Source Vehicle
7.	DS_1	Address of Destination Vehicle
8.	RT	Routing Table
9.	T_i	Time Stamp of Vehicle i
10.	HC	Hop Count
11.	PKT	Message Packet
12.	&&	Logical AND
13.	11	Logical OR

8 Simulation Results

Network Simulator 2 (NS2) is the most widely used simulator tool in academics and industry in order to perform real time analysis.

For the simulation purpose, we use Network Simulator 2 as this is the one of the most powerful simulator tool to carryout network experiments. NS2 was developed in year 2000 to analyze the performance of Congestion Control Network in TCP [31]. Still NS2 is a powerful tool used to simulate and analyze the performance of network based on various parameters *i.e.* packet loss, throughput, delay and many others. NS2 is a product of NS and it is object oriented, event driven simulator supporting C++ and TCL/OTCL languages [12, 24].

In order to detect Sybil vehicle we implement the proposed algorithm over Network Simulator 2 (NS2). For vehicle discovery and data transmission, we use Ad-Hoc On-demand Distance Vector Routing Protocol (AODV).

Algorithm 1 Working of Proposed Methodology

1: Begin
2: for i=1 to p
3: $s_i \varepsilon V$
4: **if** $V_i \varepsilon VANET(Servics)$ **then**
5: $V_i -> RSU_j(Pk_i)$
6: $RSU_j -> CA(Pk_i)$
7: **else if** $CA -> RSU_j(Pk_i) : ACK$ **then**
8: RSU_j issues Timestamp (T_i) to V_i
9: **else if** $CA -> RSU_j(Pk_i) : NAK$ **then**
10: RSU_j marks vehicle as Sybil and generates alert.
11: **end if**
12: **if** $V_i = S_0$ **then**
13: look for destination.
14: Goto Step 17.
15: **end if**
16: Repeat Step 34 while DS_1 *not found* && $HC == 0$
17: **if** $DS_1 \varepsilon RT(S_0)$ **then**
18: $PKT = Pk_i \ Ti \ DS_1 \ HC$
19: Message PKT send to DS_1
20: **else if** while (HC =! 0) **then**
21: forward request message to neighbor for DS_1
22: **else if** $V_k \ DS_1 \ || \ DS_1 \ \varepsilon \ RT(V_k)$ **then**
23: reply $PKT = Pk_iT_iDS_1HC, Pk_rT_r \cdots Pk_kT_k$
24: **else**
25: forward request message to neighbor for DS1
26: $S_i -> RSU_k(Pk_j, T_j)$
27: **end if**
28: **if** $RSU_k -> S_i(ACK)$ **then**
29: Destination Identified and Secure: Start data Transmission
30: **else**
31: $RSU_k -> S_i(NCK)$
32: Destination Identified but not Secure: Discard Packet: Forward fresh search packet for Destination
33: **end if**
34: **if** $PKT \varepsilon DS_1$ **then**
35: for $i = 1$ to $i = z$
36: **if** $(Pk\_i, \ Ti \ == \ RSU_i(Pk\_i, \ Ti))$ **then**
37: $RSU_j -> DS_1(ACK)$
38: **end if**
39: **else**
40: $RSU_j -> DS_1(NAK)$
41: **end if**
42: **if** RSU_j *send* $ACK(DS_1)$ **then**
43: goto Step 17
44: **else if** $RSU_j -> DS_1(NAK)$ **then**
45: i. Sybil vehicle is detected.
46: a. mark vehicle as Sybil vehicle.
47: b. generate alert in network.
48: ii. Request sends to destination vehicle for the retransmission of request message.
49: **end if**
50: End

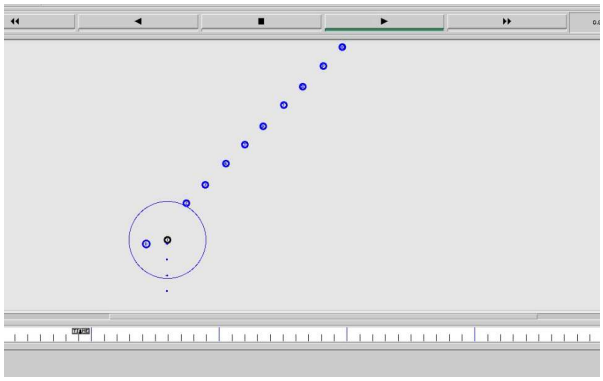


Figure 6: Detection of Sybil vehicle in VANET

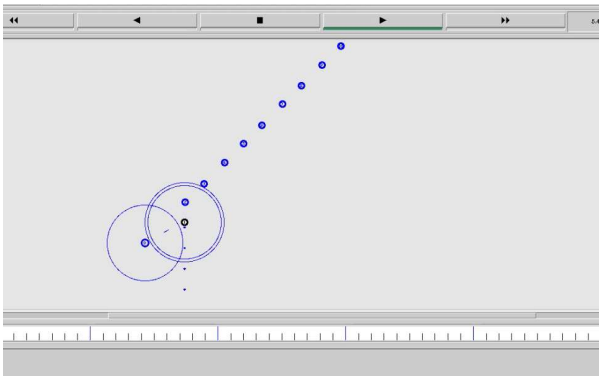


Figure 7: Detection of sybil vehicle in VANET

Where every non-falsie vehicle follows AODV protocol whereas falsie vehicle violates the principle of AODV Protocol and perform malicious behavior in network. Our work is based on real time traffic as explained earlier for that NS is most suitable simulator to implement that.

First of all we have deployed network on NS3 simulation tool. We have injected Sybil vehicle 1 in the network. We have implemented network of 15 vehicles as shown in Figure 6. We have applied AODV algorithm for routing of packets from source to destination. In VANET each vehicle has a unique address to participate in routing and there is no central authority to verify vehicles. Malicious vehicles may use different address for Route request (RREQ) and Route reply (RREP). In this work author have identified and detected vehicle 1 as malicious using proposed time stamp based algorithm in VANTE as shown in Figures 6-10.

In the proposed work RSU issued time stamp to all the vehicles and maintain a data base of time stamp issued to public key of vehicles. RSU compares the list of time stamp and public key forwarded by destination vehicle with its own time stamp issued to respective public keys. If RSU found all legal timestamp issued to public key, it replies Acknowledgement (ACK) to destination vehicle (DS_1).

Moreover, in case RSU found some illegal/wrong time stamp over corresponding public key, RSU replies Negative Acknowledgement (NAK).

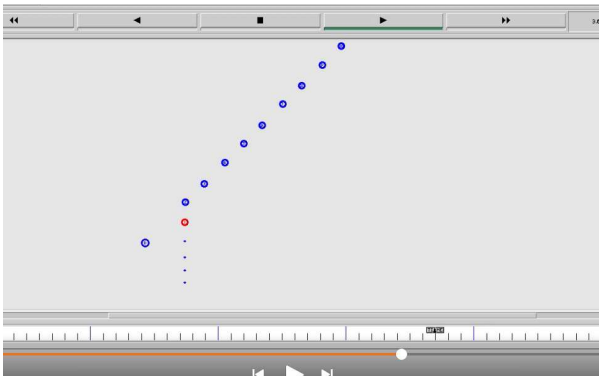


Figure 8: Detection of sybil vehicle in VANET

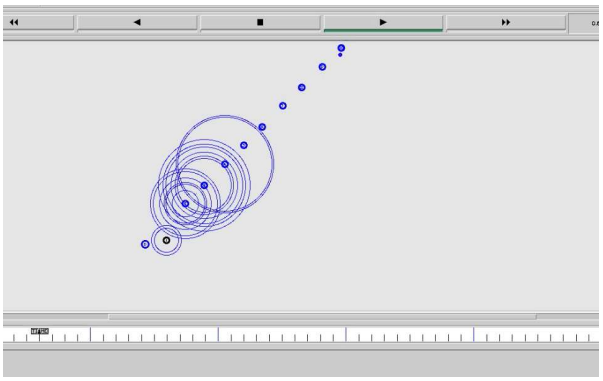


Figure 9: Detection of sybil vehicle in VANET

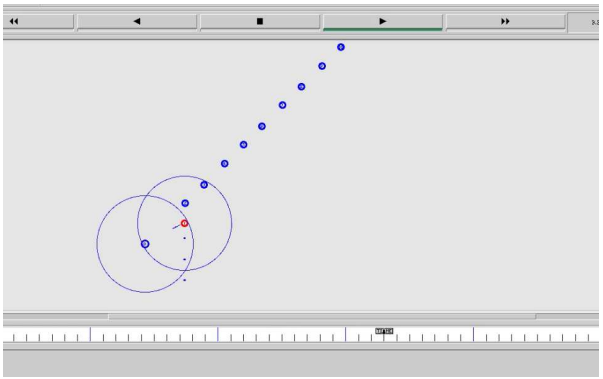


Figure 10: Detection of sybil vehicle in VANET

9 Future Work

In future work we would like to extend our work and create a network where it will be impossible to perform Sybil attack. In addition, we would like to design algorithms and examine our work for more than one Sybil vehicle and perform simulation over real time traffic model. In our future work, we will also try to build an automatic model to establish reliable relationship among components of VANET.

Acknowledgments

This study was supported by Shri Ramswaroop Memorial University of Lucknow, Uttar Pradesh, India. I would like to express my special thanks to my supervisor (Dr.Taskeen Zaidi) without her support this research paper would not be possible.

References

[1] T. W. Chim, S. Yiu, L. C. K. Hui and V. O. K. Li, "VSPN: VANET-based secure and privacy preserving navigation," in *IEEE Transactions on Computers*, vol. 63, no. 2, pp. 510–524, Feb. 2014.

[2] A. Daeinabi, A. G. Rahbar, "An advanced security scheme based on clustering and key distribution in vehicular ad-hoc networks," *Computers & Electrical Engineering*, vol. 40, no. 2, pp. 517–529, Feb. 2014.

[3] M. Faezipour, M. Nourani, A. Saeed, and S. Addepalli, "Progress and challenges in intelligent vehicle area networks," *Communications of the ACM*, vol. 55, no. 2, pp. 90–100, Feb. 2012.

[4] S. M. Faisal, A. K. Vajpayee, "Extended zone routing protocol," *International Journal of Computer Sciences and Engineering*, vol. 5, no. 5, May 2017.

[5] J. Farooq, M. S. Alouini, *et al.*, "A stochastic geometry model for multi-hop highway vehicular communication," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 2276-2291, 2016.

[6] J. Grover, M. S. Gaur, V. Laxmi and N. K. Prajapati, "A sybil attack detection approach using neighboring vehicles in VANET," in *Proceedings of the 4th International Conference on Security of Information and Networks*, pp. 151-158, 2011.

[7] J. Grover and V. Laxmi and M. S. Gaur, "Attack models and infrastructure support detection mechanism for position forging attack in vehicular adhoc networks," *CSI Transactions on ICT*, vol. 1, no. 3, pp. 261–279, Sep. 2013.

[8] S. Han, D. Ban, W. Park and M. Gerla, "Localization of sybil nodes with electro-acoustic positioning in VANETs," in *IEEE Global Communications Conference*, pp. 1-6, 2017.

[9] J. P. Hubaux, J. Luo , S. Capkun, "The security and privacy of smart vehicles," *IEEE Security & Privacy Magazine*, vol. 2, no. 3, pp. 49-55, 2004.

[10] M. Inam, Z. Li, A. Ali, and A. Zahoor, "A novel protocol for vehicle cluster formation and vehicle head selection in vehicular ad-hoc networks," *International Journal of Electronics and Information Engineering*, vol. 10, no. 2, pp. 103–119, 2019.

[11] M. Ivanov, F. Brännström, A. G. i Amat, P. Popovski, "Broadcast coded slotted ALOHA: A finite frame length analysis," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 651-662, 2016.

[12] J. P. Jeyaraj and M. Haenggi, "Reliability analysis of V2V communications on orthogonal street systems," *IEEE Global Communications Conference*, pp. 1-6, 2017.

[13] Z. Jianhong, X. Min and L. Liying, "On the security of a secure batch verification with group testing for VANET," *International Journal of Network Security*, vol. 16, no. 4, pp. 313–320, July 2014.

[14] M. Khabazian, S. Aissa and M. Mehmat-Ali, "Performance model of safety message broadcast in vehicular ad hoc network," *IEEE Transactions On Intelligent Transportation Systems*, vol. 14, no. 1, pp. 380–387, Mar. 2013.

[15] T. Kimura and H. Saito, "Theoretical interference analysis of inter-vehicular communication at intersection with power control," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'16)*, pp. 3–10, 2016.

[16] X. Liu, X. Zhang, M. Jia, L. Fan, W. Lu, X. Zhai, "5g-based green broadband com-munication system design with simultaneous wireless information and power transfer," *Physical Communication*, vol. 28, pp. 130-137, 2018.

[17] Z. Na, X. Li, X. Liu, *et al.*, "Subcarrier allocation based simultaneous wireless information and power transfer for multiuser OFDM systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, pp. 148, 2017.

[18] S. Park, B. Aslam, D. Turgut and C. C. Zou, "Defense against Sybil attack in vehicular ad hoc network based on roadside unit support," in *IEEE Military Communications Conference*, pp. 1-7, 2009.

[19] A. Rakhshan and H. Pishro-Nik, "Improving safety on highways by customizing vehicular ad hoc networks," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 2017-2026, Mar. 2017.

[20] A. Rana, D. Sharma, "Mobile ad-hoc clustering using inclusive particle swarm optimization algorithm," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 1-8, 2018.

[21] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J. P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Vehicular Networks*, vol. 25, no. 8, pp. 1557–1568, 2007.

[22] D. Sam, C. Velanganni, T. E. Evangelin, "A vehicle control system using a time synchronized Hybrid VANET to reduce road accidents caused by human error," *Vehicular Communications*, vol. 6, pp. 17-28, Oct. 2016.

[23] G. Samara, W. A. H. Al-Salihy and R. Sures, "Efficient certificate management in VANET," *The 2nd International Conference on Future Computer and Communication*, vol. 3, pp. 750-754, 2010.

[24] P. Sarkar, C. Kar, B. Sen and K. Sharma, "Sensitivity analysis on AODV with Wormhole attack," *The 2nd International Conference on Next Generation Computing Technologies (NGCT'16)*, pp. 803-807, 2016.

[25] A. K. Sharma, S. K. Saroj, S. K. Chauhan and S. K. Saini, "Sybil attack prevention and detection in vehicular ad hoc network," in *International Conference on Computing, Communication and Automation (ICCCA'16)*, pp. 594-599, 2016.

[26] P. K. Singh, S. Sharma, S. K. Nandi, S. Nandi, "Multipath TCP for V2I communication in SDN controlled small cell deployment of smart city," *Vehicular Communications*, vol. 15, pp. 1-15, 2019.

[27] A. Tassi, M. Egan, R. J. Piechocki and A. Nix, "Modeling and design of millimeter-wave networks for highway vehicular communication," in *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10676-10691, Dec. 2017.

[28] United States Department of Transportation, National Highway Traffic Safety Administration, Traffic safety facts, 2015. (`https:\//crashstats.nhtsa.dot.gov/\#/DocumentTypeList\\/11`)

[29] L. Xiao, L. J. Greenstein, N. B. Mandayam and W. Trappe, "Channel-based detection of sybil attacks in wireless networks," in *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 492-503, Sep. 2009.

[30] B. Xiao, B. Yu, and C. Gao, "Detection and localization of sybil nodes in VANETs," in *Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS'06)*, pp. 1-8, 2006.

[31] T. Zaidi and S. Faisal, "An overview: Various attacks in VANET," *The 4th International Conference on Computing Communication and Automation (ICCCA'18)*, pp. 1-6, 2018.

Biography

Syed Mohd Faisal is a research scholar in department of Computer Science and Engineering at Shri Ramswaroop Memorial University, Lucknow, India. Syed Mohd Faisal is also working as Assistant Professor in Department of Computer Application at Integral University. His area of interest is Wireless sensor Network, Computer Networks and Cloud Computing.

Dr. Taskeen Zaidi is working as an Assistant Professor in department of Computer Science and Engineering at Shri Ramswaroop Memorial University. Her area of interests is Software engg., Cloud Computing, Distributed computing and Ad-hoc networks.

Static Analysis of Superfluous Network Transmissions in Android Applications

Jianmeng Huang, Wenchao Huang, Zhaoyi Meng, Fuyou Miao, and Yan Xiong
(Corresponding author: Wenchao Huang)

School of Computer Science and Technology, University of Science and Technology of China
Elec-3 (Diansan) Building, West Campus of USTC, Huang Shan Road, Hefei, Anhui Province, China
(Email: huangwc@ustc.edu.cn)

(Received June 20, 2018; Revised and Accepted Nov. 22, 2018; First Online July 16, 2019)

Abstract

The network transmission is an important way to exchange information between Android applications and their own backend or other third-party servers. However, some network transmissions are superfluous for the apps' functionalities. Superfluous network transmissions not only increase the network traffic but also may leak users' sensitive data. To identify the superfluous network transmissions, we propose a static-analysis based approach. Evaluation with real world market apps shows that 62% apps contain superfluous network transmissions, and 48% of the analyzed network transmissions are superfluous, and our approach could effectively detect superfluous network transmissions in Android apps.

Keywords: Android Security; Privacy Leakage; Superfluous Network Transmissions

1 Introduction

In recent years, smartphone plays an important role in people's daily life. It is not simply a communication tool now, but also a data container and a personal assistant. Various functionalities of smartphones are provided by multifarious applications (apps), which can be downloaded from the app market or third parties. In 2017, the number of available apps in the Google Play Store was placed at 3 million apps [19]. As the development of Android apps, the functionalities provided by apps become more refined and personal customized. As a result, sensitive data are collected and may be transmitted via network to support these functionalities [20]. For example, an app which sells movie tickets may utilize the GPS data of the smartphone, transmitting the GPS data to remote servers and getting the recommendation of nearby cinemas.

While some network transmissions are needed to fulfill apps' functionalities, other network transmissions are superfluous. The superfluous network transmission means that the transmission is not necessary: No matter the transmission is success or not, it is of no help for the apps functionalities. Some malicious apps may intentionally leak users' privacy via network transmissions [8]. These network transmissions are superfluous and have no aid to the app functionalities. Even in benign apps, there may also be superfluous network transmissions which collect users' privacy. The superfluous network transmission does not benefit users. First, the superfluous network transmission increases the network traffic and consumes the power resource of mobile devices. Second, the superfluous network transmission may leak users' privacy. Since the transmission is not necessary for app functionalities, it is of high possibility that the transmission is useful to app providers. For example, an app provider may collect users data for advertisement purpose or user habit analysis.

Existing techniques are insufficient in detecting such superfluous network transmissions. Rubin *et al.* propose a technique [16] which focuses on detecting covert communications that have no effect on the user-observable application functionality. Its core idea is to look for cases when no information is presented to the user neither on success nor on failure of the connection. We argue that this definition about covert communication is not proper. First, covert communication could also present information to the user when the connection failure occurs, *e.g.*, when a device is put in disconnected environment or airplane mode, for that warning the user about network failure would not expose the purpose of the malicious network communication. Second, some network transmissions could also be necessary for app functionalities even though these transmissions have no direct effect on the user interface. For example, at initialization, an app may synchronize data from remote server and store it, which is not used by the app immediately but used latter by other app functionalities. Hence, it is not always appropriate to distinguish necessary network transmissions and superfluous ones by figuring out whether the transmissions (either on success or on failure of the transmission) could directly result in affecting the user interface. LeakSemantic [10]

targets for locating abnormal sensitive network transmissions from mobile apps. It consists of a program analysis component to precisely identify sensitive transmissions and a machine learning component to further differentiate between the legal and illegal network transmissions. It focuses on the sensitive data, but users' behavior data are also privacy. Besides, it uses lexical features derived from the set of URLs in the traffic traces to train classifiers, which only works for connections using `HTTP GET` request.

In this paper, we propose a novel approach to detect the superfluous network transmissions in Android apps. We model superfluous network transmissions as the transmissions of which the responses are not utilized by apps. Our key insight is that if a network transmission is necessary, the response of the network connection should be utilized by the app. Our approach decides whether a network transmission is necessary by the information of how the response of the network transmission is handled. It is different from existing researches [10, 23, 25] that detect network transmissions by figuring out how sensitive data are generated, utilized and finally transmitted out of the device. Our approach concentrates on figuring out whether the responses of network transmissions are utilized by the app. If not, we consider corresponding network transmissions as superfluous. Our approach is also different from the research [16] which detects covert communications that have no effect on the UI. We argue that only if the response of the network communication is not utilized by the app, the communications is superfluous. Overall, our approach detects superfluous network communications from a different aspect. Existing researches could be complementary to our work and they can work with our approach side by side to enhance user privacy.

To figure out whether the response is used by the app, we utilize the information flow analysis to track how the response is used. Here, we address two challenges. First, how to choose the `sources` and `sinks` of the information flow analysis. Improper `sources` and `sinks` may lead to the insufficient identification or false positives. Second, we use a novel light-weighted approach to handle the implicit data flow of the response.

Our contributions are summarized as follows.

- We study superfluous network communications and propose a new way of distinguishing superfluous and necessary network communications.
- We propose a static analysis approach which automatically detects superfluous network transmissions in Android apps.
- We demonstrate the effectiveness of our approach with real-world Android applications.

The rest of this paper is organized as follows. Section 2 introduces the background and states the problem. Section 3 details the design of our approach and after that, Section 4 describes experimental results. Section 5 discusses the future work of our approach and Section 6 describes related work. Finally, Section 7 concludes the paper.

2 Background and Problem Statement

2.1 Network Transmissions in Android Apps

In order to perform network operations in Android applications, many APIs are developed. Table 1 lists the base classes and methods which are responsible for network connections and data transmissions. The third column of the table lists the methods which are responsible for getting the return value of network connections. Most network-connected Android apps use HTTP to send and receive data. Besides, developers could also use other basic JAVA network connecting APIs, which is listed in the last four rows in the table.

Developers should use an asynchronous task for network transmissions so the UI thread doesn't freeze. If the UI thread freezes, Android will show an "Application not responding" dialog to the user. To avoid creating an unresponsive UI, it is recommended not to perform network operations on the UI thread. By default, Android 3.0 and higher versions require apps to perform network operations on a thread other than the main UI thread; if not, a `NetworkOnMainThreadException` is thrown. To facilitate the deployment of network transmission, many third-party libraries provide APIs to encapsulate asynchronous network operations. Generally, these libraries are implemented based on the base classes and methods.

In some network transmission libraries, using asynchronous requests forces developers to implement a `Callback` with its two callback methods: success and failure (*i.e.*, `onResponse` and `onFailure()`). When calling the asynchronous `getTasks()` method from a service class, developers have to implement a new `Callback` and define what should be done once the request finishes.

2.2 Problem Statement

```
1 public String sendData(){
2   String message;
3   try {
4     OkHttpClient client = new OkHttpClient
          ();
5     FormBody.Builder formBody = new
          FormBody.Builder();
6     formBody.add(``username'',``foo'');
7     Request request = new Request.Builder
          ()
8       .url("http://www.sample.com")
9       .post(formBody.build())
10      .build();
11    Response response = client.newCall(
          request).execute();
12  // if (response.isSuccessful()) {
13  //   message = response.message();
14  // }
15   } catch (Exception e) {
16     e.printStackTrace();
17   }
```

Table 1: The considered network connection APIs

Class or Interface	Connecting	Getting Response
java.net.URLConnection	Connect	GetInputStream
java.net.URL	OpenConnection	OpenStream
org.apache.http. client.HttpClient	Execute	Execute
java.net.Socket	GetInputStream getOutputStream	GetInputStream
com.squareup. okhttp.OkHttpClient	NewCall	NewCall

```
18   return message;
19 }
```

Listing 1: An example of network transmission

As a motivating example, Listing 1 shows an example of superfluous network transmission. The network transmission is implemented based on the `okhttp`. It sends the keyword *username* to the remote server by HTTP POST request. If the response of the network connection is not used by the app (*i.e.*, the codes commented from line 12 to line 14), the function of method `sendData` is simply sending data to the remote server. This network transmission may be useful for the app provider, but it is of no use for the app's functionalities for that the app does not gain any feedback from the network transmission.

To handle the problem, we face the challenge of deciding whether the response of a network transmission is utilized by the app. First, we should trace how the response is used and figure out whether it is used for necessary app functionalities. Second, there may be implicit information flow using the response, which increases the difficulty of tracing the use of the response.

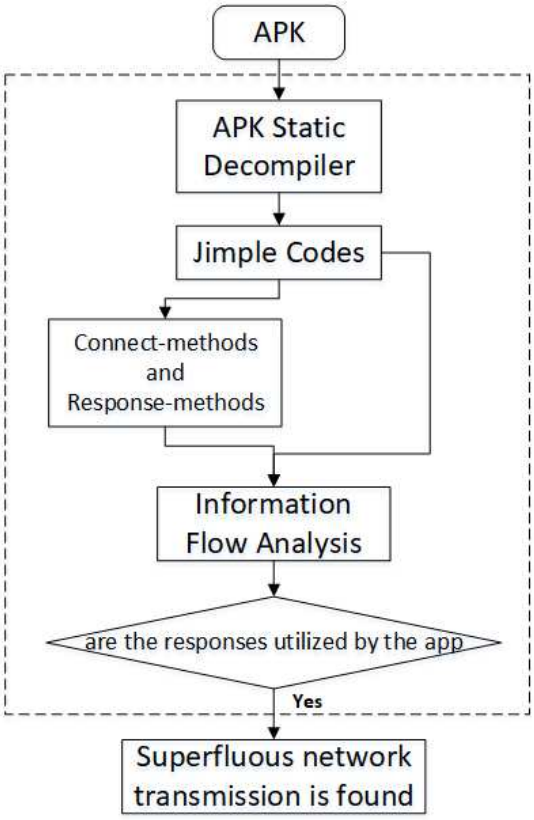


Figure 1: Work-flow of our approach

3 Design and Implement

We propose a static analysis approach to find the superfluous network transmissions in Android applications. This section describes how we model the superfluous network transmissions and how we identify them in Android apps.

3.1 Overview

The core idea behind our approach is to look for cases where the responses of network transmissions are not utilized by apps. We determine whether a network transmission is superfluous by tracking how the response is used by the app. If the response is not utilized by the app or improperly used, we deem the network transmission superfluous.

Guided by this idea, Figure 1 shows the work-flow of our approach. Given an app, we first translate the apk file of it into an intermediate representation (*e.g.*, Jimple representation, a statement based intermediate representation), based on which we could apply static analysis. This process is commonly used by static analysis for Android apps [1, 4, 22]. Then we search the code segments which are responsible for network transmissions in the intermediate representation. Afterwards, we apply information flow analysis to check whether the responses of network transmissions are utilized by app functionalities. Here, the information flow analysis tracks how sensitive information is propagated through an application. Finally, we find the superfluous network transmissions by identifying unused responses.

To apply the information flow analysis, we use FlowDroid [1] as the underlying analysis infrastructure. FlowDroid provides taint analysis which presents potentially data flows to human analysts or to automated app-detection tools which can then decide whether a data use actually constitutes a policy violation. It adequately models Android-specific challenges like the application lifecycle or callback methods, which helps reduce missed leaks or false positives. Our analysis is based on information flow analysis of FlowDroid on Android apps. Information flow analysis is a common technique used for analyzing Android apps in that it can track how data are potentially

be used by the app, which helps to distinguish different use of the data (*e.g.*, improper use and legitimate use).

We use static analysis based approach because it is usually more efficient [13]. Static analysis is applied without executing the code. It relies on Java bytecode extracted by disassembling an application. As a result, static analysis could help analyst to inspect all the behaviors of an app. We do not adopt dynamic analysis approach because dynamic analysis faces the problem of low testing coverage, which causes insufficient analysis of the app. In dynamic analysis, analysts have to execute (or emulate) the app in order to collect runtime information used for further analysis. However, it cannot be guaranteed that all the code paths in an app are executed during the dynamic analysis, because some code paths require particular trigger conditions. For example, for a shopping app, an app behavior may only occur when the user buys something. Besides, dynamic analysis executes only one code path of an app at one time, while static analysis could be applied in parallel. Overall, compared with static analysis, it is not efficient to test apps with dynamic analysis.

3.2 Design

In this subsection, we describe how we find the superfluous network transmissions. We first introduce cases which are considered to be superfluous network transmissions. Then we describe how our static analysis approach handles these cases.

3.2.1 Models of Superfluous Network Transmissions

To automatically detect superfluous network transmissions in Android apps, we first study network transmissions in malicious and benign Android apps to find the superfluous ones and draw the common points of them. We also study the different features between legitimate network transmissions and superfluous ones. Particularly, we denote the *connect method* as method which is responsible for creating connections between devices and remote servers, and we denote *response method* as method which is responsible for getting responses from the connection. Table 1 shows the *connect method*s and *response method*s monitored by our approach. Overall, we summarize the following three categories as superfluous network transmissions considered in this paper.

Category 1: A network transmission calls the *connect method* but does not call the *response method*. In this category, the network connection is established, but the response of the connection is not handled. As a result, the transmission simply sends data to remote server. The app is not going to establish the connection again if the transmission fails. Usually, for a necessary network transmission, the app would at least query the status of the transmission (*e.g.*, querying the HTTP status code) to check if the network transmission fails. If it fails, the app would handle the failure in an exception and then inform users that the network is not available. Network transmissions in this category are deemed superfluous for that they are not properly deployed and the purposes of these transmissions are not clear. These superfluous network transmissions may be caused by careless app developing or malicious privacy collecting which avoids being noticed by users.

Category 2: A network transmission calls both *connect method* and *response method*, but the app does not actually utilize the response of the network transmission. At the code level, the *response method* is invoked, but the value of the response is not passed to other codes (*i.e.*, `method`s developed by the app) in the app. In this category, although the app gets the response of the network transmission, it does not make use of the response for the app functionalities. Hence, the network transmission does not contribute to the normal app functionalities. Network transmissions in this category may be caused by the iterative development of the app. Some functionalities of the app are discarded but the corresponding codes are not clearly removed.

Category 3: A network transmission calls *connect method* more than one time, and the URLs of the connection are different. We find that some apps send the same data to multi servers simultaneously. One of the transmission addresses belongs to the app provider, but other ones are data centers. We consider that at least one of the network transmissions is superfluous, because the transmitted data are the same and one response from the remote server is enough for the app functionalities. The purpose of network transmissions in this category is that one network transmission is used for necessary app functionalities, and others are used for transmitting users' data to data centers, which consumes the network resources of mobile devices. Note that if the network transmissions are in different branch statements, we do not consider them as superfluous ones. Because the app provider may own multi servers, some of which are alternate servers. Hence, in case the main server is down, the network transmissions in different branches could be established.

3.2.2 Finding Superfluous Network Transmissions

We further develop a static analysis approach to identify superfluous network transmissions in the above categories. To handle the three categories of superfluous transmissions, we develop an algorithm based on information flow analysis. Information flow analysis tracks sensitive "tainted" information through the app by starting at a pre-defined source (*e.g.*, an API method returning the response of a network transmission) and then following the data flow until it reaches a given sink (*e.g.* a method

Algorithm 1 Identifying superfluous network transmissions

```
1:  Input: The map of connect method and response method methodMap, and a method m in the app
2:  Output: Whether the app has superfluous network transmissions
3:  Begin
4:  Count ⇐ times of methodMap.key() invoked by m
5:  if count > 1 then
6:    if sending same data to different addresses then
7:      return true
8:    end if
9:  end if
10: if count > 0 then
11:   for all connect_method invoked by m do
12:     response_method ⇐
        methodMap[connect_method]
13:     if response_method is also invoked then
14:       response ⇐
          the result of methodMap[connect_method]
15:       if response is not propagated out of m or the
          callback method then
16:         return true
17:       end if
18:     else
19:       return true
20:     end if
21:   end for
22: end if
23: return false
24: End
```

writing the information to a UI element), giving precise information about which data may be leaked.

Algorithm 1 shows how we identify superfluous network transmissions. It accepts the maps of *connect method* and *response method*, and a method m in the app. Here, the maps of *connect method* and *response method* are pairs of methods of the same network transmission library, and the overloaded methods are also included in the maps. The output of this algorithm is a judgment about whether the app contains superfluous network transmissions. We first decompile the apk file into bytecodes, then we utilize Soot [21] to translate the bytecodes to intermediate representation (*i.e.*, Jimple). Based on the Jimple representation, we get all the `methods` defined by the `classes` in the app. Then, we look for superfluous network transmissions in each method using Algorithm 1.

For each method m in the app, we look for network transmissions and then figure out whether the transmissions are superfluous. Figure 2 illustrates the methodology of Algorithm 1. It first checks superfluous network transmissions in category 3. It figures how many times the *connect methods* are invoked in the method m. If the *connect methods* are invoked more than one times, which is the situation in category 3, we check whether there exist more than two network transmissions send the same data

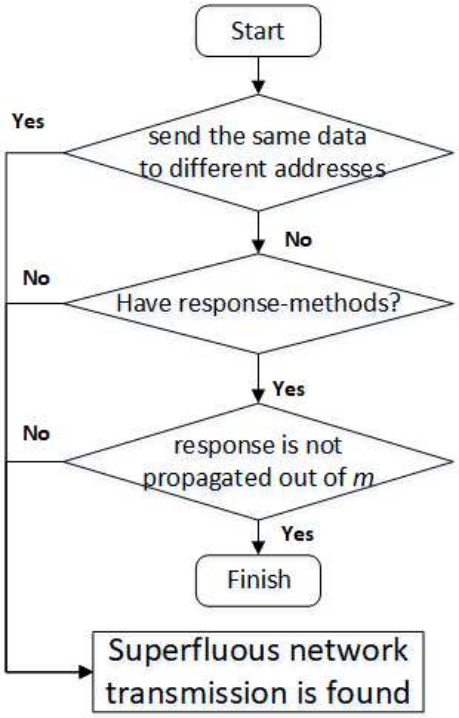


Figure 2: Methodology of identifying superfluous network transmissions

to different addresses. If found, we report the superfluous network transmissions (*i.e.* line 6 to 8).

Then, Algorithm 1 checks superfluous network transmissions in category 1, as shown by Figure 2. If *connect methods* are invoked, Algorithm 1 would check each of the *connect methods* (*i.e.*, line 11). Then for each *connect method*, we check whether the corresponding *response method* is called in the method m or whether there is a corresponding callback method (*e.g.*, `onResponse`), which is designed for asynchronous purpose. If not, we report the connection as superfluous one.

Otherwise, Algorithm 1 checks whether the result of the *response method* is utilized by the app (*i.e.*, line 13 to line 17). The process is illustrated by Figure 2. Here, the information flow analysis is used to handle the situation in category 2. Specially, we set the results of the *response methods* as *sources* and the `return` statement of the method as the *sink*. Besides the *response methods*, we also include other APIs, which utilizes the response of network connection, in the *sources*. *e.g.*, HttpURLConnection.getResponseCode(), HttpResponse.getStatusLine(), HttpResponse.getEntity(). As a result, if the response is propagated via the data flow out of the method m or the callback method, which indicates that the response would be utilized by other methods, this network transmission is deemed necessary. Otherwise, the connection is deemed superfluous.

We set the *sources* and *sinks* in two configuration files, which enables the scalability of our approach. As a result, if we find new third party libraries which are responsible for network transmissions, we can add the corresponding

methods into our configuration files.

Additionally, we handle the implicit data flow of utilizing responses of network transmissions. Listing 2 shows an example of returning the status of a network transmission. The `response` is used to get the *status code* of the network transmission (*i.e.*, line 6). The `AllSuccess` indicates that whether the connection succeed, but there is no explicit data flow from the `response` to the `AllSuccess`. To handle this situation, we could take advantage of existing approaches, such as EdgeMiner [4], which addresses implicit flows in static analysis. As we only need to handle implicit data flow for specific resources (*e.g.*, `getStatusCode()`), we use a lightweight way to handle this problem.

```
1 public boolean sendData(){
2   boolean allSuccess = true;
3   try {
4     ...
5     HttpResponse response = client.execute
          (http, httpContext);
6     if (response.getStatusLine().
          getStatusCode() != 204) {
7       allSuccess = false;
8     }
9   } catch (Exception e) {
10    e.printStackTrace();
11  }
12  return allSuccess;
13 }
```

Listing 2: Implicit data flow of utilizing response

Our lightweight way specifically monitors the branch statements to address the implicit data flow of using `response`. In detail, we first locate the methods which contain network transmissions in the app. Then we locate the branch statements in these methods. If the conditional statement of the branch calls the APIs which utilize the response of the network connection, we would trace the variables in the branches. Finally, if none of these variable contributes to the variable which is to be returned by the method, we consider this network transmission as superfluous.

4 Evaluation

In this section, we intend to evaluate our approach in the following aspects. First, how effective is our approach in identifying superfluous network transmissions? Second, how often does superfluous network transmission occur in real-world applications?

4.1 Real World Apps Study

We first apply our approach to real-world apps in order to assess its effectiveness. We analyze 12 apps with our approach and manually check the results. As there are no researches or reports about superfluous network transmissions in Android apps, we need to manually inspect the source codes to get the ground truth of the tested apps. Hence, we choose four open-source apps from F-Droid [7], an installable catalogue of free and open source applications for the Android platform. To include commercial apps, other four apps are from wandoujia, a popular third party Android app market, and Google Play. The experiments are conducted on a 4-processor 16GB-RAM machine.

The names and package names of apps, as well as the analysis results, are listed in Table 2. Given the ground truth information (*i.e.*, the results from manual inspecting) and the analysis results, there are four possible outcomes: True positive (TP), true negative (TN), false positive (FP) and false negative (FN). TP means that an app contains superfluous network transmissions with respect to ground truth and our approach detects the superfluous transmissions. TN means that an app does not contain superfluous network transmissions with respect to ground truth and our approach does not find superfluous network transmissions in the app. FP and FN have similar meanings. The metric accuracy is computed by the following formulas:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

For open source apps, our approach finds all the network transmissions, and it detects no superfluous network transmissions in these apps. The results are manually checked by reviewing the source codes. As these apps are open-source, the functionalities of these apps are explicit. If there exist superfluous network transmissions which may leak users' privacy in these apps, the developers would be blamed. Thus there are no superfluous network transmissions which have no use for the app functionalities. As listed in the last column in Table 2, we do not find superfluous network transmissions in these apps both with our approach or manually checking the source codes. The analysis results on open source apps show that our approach has no false positives. Besides, the analysis results have no false negatives. As a result, we get 100% accuracy of analyzing open source apps.

For the market apps, after our automatically analysis, we also manually check the results in the decompiled codes. As these commercial apps does not provide the source codes of them, we utilize dex2jar and jd-gui to get the decompiled Java codes and retrieval the detected results in the decompiled codes. The first 4 apps are from wandoujia market and the last 4 apps are downloaded from Google Play. The results, listed in Table 2, show that our approach successfully finds all the network transmissions in the app and precisely identifies superfluous network transmissions of the three categories. The accuracy of the analysis result is also 100%. Overall, experimental results of the 8 apps demonstrate the accuracy of our approach on commercial apps.

Our experiments also show that the time overhead of our static analysis is acceptable, regarding to the size of each app. Similar to approaches based on information flow analysis [1, 11], it is time-consuming to analyze the app when the size of the app is large, because the method

Table 2: Results of app study (M: manually checked; A: automatically detected)

App sources	App Name	Size	Time	Network transmissions (M/A)	Superfluous (M/A)
Open source	Battery Dog	22K	3s	0/0	0/0
	ArchWiki Viewer	1.1M	6s	1/1	0/0
	Commons	20M	138s	3/3	0/0
	External IP	9.9K	3s	1/1	0/0
App market	zuimei weather	18M	514s	7/7	4/4
	sogou novel	12M	220s	18/18	11/11
	Kugou Music	47M	1672s	10/10	4/4
	karaoke	42M	941s	18/18	6/6
	Vault-Hide SMS	11M	244s	12/12	4/4
	Duolingo	21M	518s	8/8	2/2
	ibis Paint X	31M	729s	9/9	2/2
	Magzter	16M	331s	31/31	12/12

invocation relations in the app become complicated. In our experiments, we observe that the open source apps have smaller sizes than market apps, and the time overhead is lower. The reason may be that most market apps are obfuscated to avoid code plagiarism and vulnerability searching [3]. The released apps are more complexed after obfuscation, increasing the analysis time overhead. Unlike the approach [16] which identifies a network transmission by judging whether the result of the transmission has direct effect on the user interface, our approach adopts some simplifications to improve scalability, such as judging whether the response of a network transmission is utilized by the app by tracing the data flow of the response until the `return` statement of the current method. As a result , we demonstrate that the our static analysis achieves relatively high performance.

Furthermore, we analyze 100 apps downloaded from wandoujia and 100 apps from Google play to figure out how common is the superfluous network transmissions in real world apps. The analyzed apps are the most popular apps in the market collected from different categories such as games, tools, entertainment, weather, social and sports. Experiments on these apps show that for the apps from wandoujia market, 62% of the apps contain superfluous network transmissions. Besides, of all the network transmissions in these apps, 48% of them are identified as superfluous network transmissions by our app. For the apps from Google Play, 22% of the apps contain superfluous network transmissions, and 43% of the network transmissions are identified as superfluous ones. Overall, we can conclude that superfluous network transmissions exist in real world apps with high proportion.

4.2 Finding and Case Study

We find that most of the superfluous network transmissions are collecting users' data to remote servers. Most of the transmitted data are related to user's identity (*e.g.*, device id, product id, IMEI, *etc.*). Besides, the superfluous network transmissions are often triggered by UI elements which are frequently triggered. We can conclude that collecting user's personal data are an important purpose of superfluous network transmissions. For example, *KugouMusic* is a popular music app in the app market. Our approach finds that it has 4 superfluous network transmissions. We decompile the apk file and locate the code segments of the superfluous network transmission. Then we trace back to the event which leads to the superfluous network transmission: as shown in Figure 3, the *Watch* button would lead to the network transmission. In users' expectation, the *Watch* button should only provide the function of switching between different UIs. It should not lead to any network transmission. Hence, this network transmission here is superfluous for the app functionality. Furthermore, we inspect the `address` of the superfluous network transmission at runtime, we find the address belongs an Internet Data Center (IDC) provider in Beijing rather than the command and control server. Hence, we believe that our insight of detecting superfluous network transmission is reasonable and useful.

Figure 3: The GUI of KugouMusic

We also find that blocking the superfluous network transmission would not impact the app functionalities. We manually disable the detected superfluous network transmission in the tested apps and repackage the apps. Then we compare the app functionalities between the original apps and repackaged apps. For the two versions of each app, we feed them with the same inputs. If the

user interfaces which represent the functionalities are different during the test, *e.g.*, the repackaged app crashes or some information in the user interfaces of the repackaged app is missing, the corresponding app functionality are impacted. Finally, we observe that the blocked superfluous network transmissions have no impact on the normal app functionalities.

5 Discussion

After identifying the superfluous network transmissions in apps, the results could be reported to app markets or app providers. Our approach can be used by app markets to display the detection result of each app, which urges the app provider to make a clear statement about how the app would use users' private data in the privacy policy (listed in the app market). The app could also prompt a privacy collecting request to users. If users do not agree the app to collect their private data, the app should not transmit private data to remote servers. Besides, our approach can be adopted by app providers to check whether the identified superfluous network transmissions are caused by careless coding, which helps to improve the code developing of the app.

The superfluous network transmissions can be blocked to reduce the risk of privacy leakage if the app provider does not provide revised version of the app. There are two possible solutions to automatically block the detected superfluous network transmissions. The first one is to statically disable the code segments which are responsible for superfluous network transmissions. Similar to prior researches [5, 12, 17], we could reduce the unwanted code segments which are responsible for superfluous network transmissions, and then repackage the app. As a result, the repackaged app does not contain superfluous network transmissions. The second solution is to record the patterns of the superfluous network transmissions, and then disable the transmissions at runtime, which could take advantage of the framework of a prior approach [14]. Blocking the superfluous network transmissions is beyond the research scope of this paper and we leave it as our future work.

6 Related Work

There are several approaches to analyze the behaviors of Android apps. FlowDroid [1] and DroidSafe [11] provide static taint-analysis tools to detect potentially malicious data flow in Android applications. Our approach utilizes the data flow analysis provided by FlowDroid. TaintDroid [6] and TaintART [18] propose system-wide information flow tracking tool that can simultaneously track multiple sources of sensitive data. They are dynamic-based approaches and hence face the problem of low test coverage. Amandroid [22] presents a general static analysis framework for security analysis of Android applications. It can precisely track the control and data flow of an app across multiple components, and can compute an abstraction of the app's behavior in the forms of an inter-component data-flow graph and data dependence graph. However, high privacy requires more time and computing resources. Qian *et al.* [15] combine static and dynamic techniques to find potential risks in an app and then embed monitoring code in the app. As a result, their approach could report the content of data transmissions when users are running the app. However, it is not efficient because it relies on users' help to decide whether the application leaks users' privacy. Zhao *et al.* [26] detect Android malwares based on the idea that most of the malware variants are created using automatic tools. Their approach statically extracts necessary features from each app and uses convolutional neural network to identify malwares, but it is not target for newly released malwares.

To reveal data leaks in apps and protect users' privacy, AppAudit [24] comprises a static API analysis that can effectively narrow down analysis scope and an innovative dynamic analysis which could efficiently execute application bytecode to prune false positive and confirm data leaks. AppIntent [25] detects the improper behavior that when a data transmission is not intended by the user, it is more likely a privacy leakage. It helps analysts to determine whether a data transmission is user-intended or not by providing a corresponding sequence of GUI manipulations. Apposcopy [9] presents a semantics-based approach for identifying a prevalent class of Android malware that steals private user information. MUDFLOW [2] learns "normal" flows of sensitive data from trusted applications to detect "abnormal" flows in possibly malicious applications. Leaksemantic [10] identifies suspicious sensitive network transmissions from mobile apps automatically. It utilizes machine learning classifiers to differentiate among the disclosures based on features derived from URLs in the traffic traces. These approaches focus on detecting data transmissions which are malicious or not intended by users, and they concentrate on revealing the process of data transmissions. Rubin *et al.* propose a technique [16] which focuses on detecting covert communications that no information is presented to the user neither on success nor on failure of the connection. In our work, we detect superfluous network transmissions by investigating the responses of network transmissions. Our approach studies the features of network communications and concludes the common points of how apps handle responses of superfluous network transmissions. Then we utilize the static information flow analysis to identify the superfluous network communications of which the responses are not used by the app. Overall, our work can be used as a complementary with existing researches.

7 Conclusion

The network transmission is an important way to exchange information between Android apps and remote

servers for user required app functionalities. However, it is also used by improper behaviors to leak users' privacy. We propose a novel solution to detect superfluous network transmissions in Android applications. We take advantage of static information flow analysis to track how the responses of network transmissions are used by apps. The network transmissions are deemed superfluous if their responses are not utilized by apps. Our experimental results show that superfluous network transmissions are commonly existed in Android apps, and our approach can effectively detect superfluous network transmissions in Android apps.

Acknowledgments

The research is supported by National Natural Science Foundation of China under Grant No.61572453, No.61202404, No.61520106007, No.61170233, No.61232018, No.61572454, Natural Science in Colleges and Universities in Anhui Province under Grant No.KJ2015A257, and Anhui Provincial Natural Science Foundation under Grant No.1508085SQF215. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. L. Traon, D. Octeau, and P. McDaniel, "Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps," *ACM Digital Library*, vol. 49, no. 6, pp. 259–269, 2014.

[2] V. Avdiienko, K. Kuznetsov, A. Gorla, A. Zeller, S. Arzt, S. Rasthofer, and E. Bodden, "Mining apps for abnormal usage of sensitive data," in *Proceedings of the 37th International Conference on Software Engineering*, pp. 426–436, 2015.

[3] B. Bichsel, V. Raychev, P. Tsankov, and M. Vechev, "Statistical deobfuscation of android applications," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 343–355, 2016.

[4] Y. Cao, Y. Fratantonio, A. Bianchi, M. Egele, C. Kruegel, G. Vigna, and Y. Chen, "Edgeminer: Automatically detecting implicit control flow transitions through the android framework," in *NDSS*, TR-UCSB-2014-05, 2015.

[5] J. Cito, J. Rubin, P. Stanley-Marbell, and M. Rinard, "Battery-aware transformations in mobile applications," in *The 31st IEEE/ACM International Conference on Automated Software Engineering (ASE'16)*, pp. 702–707, 2016.

[6] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B. G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones," *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, pp. 393-407, 2010.

[7] F-Droid, *F-Droid - Free and Open Source Android App Repository*, 2010-2018. (`https://f-droid.org/en/`)

[8] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, M. Conti, and M. Rajarajan, "Android security: A survey of issues, malware penetration, and defenses," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 998–1022, 2015.

[9] Y. Feng, S. Anand, I. Dillig, and A. Aiken, "Apposcopy: Semantics-based detection of android malware through static analysis," in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pp. 576–587, 2014.

[10] H. Fu, Z. Zheng, S. Bose, M. Bishop, and P. Mohapatra, "Leaksemantic: Identifying abnormal sensitive network transmissions in mobile applications," in *IEEE Conference on Computer Communications, IEEE*, pp. 1–9, 2017.

[11] M. I. Gordon, D. Kim, J. H. Perkins, L. Gilham, N. Nguyen, and M. C. Rinard, "Information flow analysis of android applications in droidsafe," in *NDSS*, 2015. (`https://people.csail.mit.edu/rinard/paper/ndss15.droidsafe.pdf`)

[12] J. Huang, Y. Aafer, D. Perry, X. Zhang, and C. Tian, "Ui driven android application reduction," in *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*, pp. 286–296, 2017.

[13] A. Kapratwar, *Static and Dynamic Analysis for Android Malware Detection*, 2016. (`https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1488&context=etd_projects`)

[14] Y. K. Lee, J. Y. Bang, G. Safi, A. Shahbazian, Y. Zhao, and N. Medvidovic, "A sealant for inter-app security holes in android," in *Proceedings of the 39th International Conference on Software Engineering (ICSE'17)*,, pp. 312–323, 2017.

[15] Q. Qian, J. Cai, M. Xie, and R. Zhang, "Malicious behavior analysis for android applications," *International Journal of Network Security*, vol. 18, no. 1, pp. 182–192, 2016.

[16] J. Rubin, M. I. Gordon, N. Nguyen, and M. Rinard, "Covert communication in mobile applications (t)," in *The 30th IEEE/ACM International Conference on Automated Software Engineering (ASE'15)*, pp. 647–657, 2015.

[17] J. Seo, D. Kim, D. Cho, I. Shin, and T. Kim, "Flexdroid: Enforcing in-app privilege separation in android," in *NDSS*, 2016. (`https://gts3.org/assets/papers/2016/seo:flexdroid.pdf`)

[18] M. Sun, T. Wei, and J. Lui, "TaintART: A practical multi-level information-flow tracking system for an-

droid runtime," in *Proceedings of ACM*, pp. 331-342, 2016. ISBN: 978-1-4503-4139-4.

[19] Tableau Software, *Number of Google Play Store Apps 2017 — Statistic*, 2017. (`https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/`)

[20] E. Toch, Y. Wang, and L. F. Cranor, "Personalization and privacy: A survey of privacy risks and remedies in personalization-based systems," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1-2, pp. 203–220, 2012.

[21] R. Vallée-Rai, P. Co, E. Gagnon, L. Hendren, P. Lam, and V. Sundaresan, "Soot-a java bytecode optimization framework," in *Proceedings of the Conference of the Centre for Advanced Studies on Collaborative Research*, pp. 13, 1999.

[22] F. Wei, S. Roy, X. Ou, and Robby, "Amandroid: A precise and general inter-component data flow analysis framework for security vetting of android apps," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 1329–1341, 2014.

[23] S. Wu, P. Wang, X. Li, and Y. Zhang, "Effective detection of android malware based on the usage of data flow apis and machine learning," *Information and Software Technology*, vol. 75, pp. 17–25, 2016.

[24] M. Xia, L. Gong, Y. Lyu, Z. Qi, and X. Liu, "Effective real-time android application auditing," in *IEEE S&P*, 2015. ISBN: 978-1-4673-6949-7.

[25] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang, "Appintent: Analyzing sensitive data transmission in android for privacy leakage detection," in *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security*, pp. 1043–1054, 2013.

[26] Y. Zhao and Q. Qian, "Android malware identification through visual exploration of disassembly files," *International Journal of Network Security*, vol. 20, no. 6, pp. 1061–1073, 2018.

Biography

Jianmeng Huang received the B.S. degree in computer science from University of Science and Technology of China in 2013. He is currently working towards the Ph.D. degree at the Department of Computer Science and Technology, University of Science and Technology of China. His current research interests include information security and mobile computing.

Wenchao Huang received the B.S. and Ph.D degrees in computer science from University of Science and Technology of China in 2006 and 2011, respectively. He is an associate professor in School of Computer Science and Technology, University of Science and Technology of China. His current research interests include information security, trusted computing, formal methods and mobile computing.

Fuyou Miao received his Ph.D of computer science from University of Science and Technology of China in 2003. He is an associate professor in the School of Computer Science and Technology, University of Science and Technology of China. His research interests include applied cryptography, trusted computing and mobile computing.

Yan Xiong received the B.S., M.S., and Ph.D degrees from University of Science and Technology of China in 1983, 1986 and 1990 respectively. He is a professor in School of Computer Science and Technology, University of Science and Technology of China. His main research interests include distributed processing, mobile computing, computer network and information security.

Improved Elliptic Curve Cryptography with Homomorphic Encryption for Medical Image Encryption

Shoulin Yin, Jie Liu, and Lin Teng
(Corresponding author: Jie Liu)

Software College, Shenyang Normal University[3]
Shenyang 110034, China
(Email: ljnan127@163.com)

(Received July 26, 2018; Revised and Accepted Dec. 20, 2018; First Online July 8, 2019)

Abstract

Medical image contains sensitive information of patients. In order to improve the efficiency and security of medical image encryption, we propose an improved elliptic curve cryptography by combining with homomorphic encryption in this paper. Traditional elliptic curve cryptography has some disadvantages, so we first make improvement for elliptic curve cryptography. Then the modified elliptic curve cryptography combining with homomorphic encryption is used in the process of medical image encryption. The experimental results show that compared with other algorithms, this new algorithm not only has good encryption effect, high security and large amount of key, but also has good sensitivity to initial value and anti-attack ability.

Keywords: Elliptic Curve Cryptography; Homomorphic Encryption; Medical Image Encryption

1 Introduction

With the rapid development of computer technology and multimedia technology, multimedia communication has gradually become an important way for people communicating with each other. At the same time, when people use computer communication to contact each other, a new request is put forward when confidential information transforming in network. Information security has gradually become the research focus. In the multimedia information, vivid image information has become one of the important means for human to express information, when it refers to confidential image information such as military, business and industry, information must be encrypted then it can transfer in the Internet [8, 10, 19].

Image encryption technology currently has the following three types:

1) Based on modern cryptography [20, 21]. Both commercially and militarily widely use the modern cryptography. In technically, image information as a data format is fully capable of being encrypted by modern cryptography including symmetric cryptography and asymmetric cryptography. In practical applications, symmetric cryptography is mainly used to encrypt commercial or military information, it is often used to encrypt short messages.

2) Based on image pixel scrambling [14, 16]. The represented approaches are Arnold transform and the magic square transform. These encryption algorithms directly act on the pixels of the image. According to some linear transformation, it changes the position of the pixel to achieve the purpose of image encryption.

3) Based on chaotic technique [2, 13]. due to the development of the chaotic dynamics in recent years, people gradually realize that the chaos can be used as a new password system, which can be used to encrypt text voice and image data. Chaos is used as a new cryptosystem which is determined by the properties of chaotic system itself.

For image encryption, there are some discoveries. Mccarthy [11] discussed that an identity-based encryption scheme enabled the efficient distribution of keys in a multi-user system. Such schemes were particularly attractive in resource constrained environments where critical resources such as processing power, memory and bandwidth were severely limited. This research examined the first pragmatic lattice-based IBE scheme and brought it into the realm of practicality for using on small devices. Assad [1] proposed a new fast, simple, and robust chaos-based cryptosystem structure and analyzed its performances. The cryptosystem used a diffusion layer followed by a bit-permutation layer, instead of byte-permutation, to shuffle the positions of the image pixels. Moreover, the

permutation layer was achieved by a new proposed formulation of the 2D cat map that allowed an efficient implementation, measured by the time complexity, in terms of arithmetic and logic operations, and also, in terms of clock cycles, of the key-dependent permutation process in comparison with the standard one. Hariyanto [4] presented arnold's cat map algorithm in digital image encryption. Su [15] proposed an image encryption scheme based on chaos system combining with DNA coding and information entropy, in which chaos system and DNA operation were used to perform substitution, and entropy driven chaos system was used to perform permutation. However, two vulnerabilities were found and presented in this paper, which made the encryption fail under chosen-plaintext attack. A complete chosen-plaintext attack algorithm was given to rebuild chaos systems' outputs and recover plain image, and its efficiency was demonstrated by analysis and experiments.

So this paper proposes an improved elliptic curve cryptography by combining with homomorphic encryption for medical image encryption. The rests of the paper are organized as follows. Section 2 introduces the improved elliptic curve cryptography. New medical image encryption is illustrated in Section 3. Section 4 outlines the experiments. Section 5 finally concludes the paper.

2 Improved Elliptic Curve Cryptography

2.1 Elliptic Curve Cryptography

Assuming that user A wants to send the encrypted plaintext m to B. A needs to execute the following operation [6]:

1) User A selects one elliptic curve E and one point in E as base point G.
2) User A selects private k and produces a public key $K = kG$.
3) User A sends E, G and public key K to user B.
4) User B receives this message, it codes the plaintext to one point M in E and randomly generates integer $r(r < n)$.
5) User B calculates $C1 = M + rK$ and $C2 = rG$.
6) User B sends $C1$ and $C2$ to user A.
7) User A receives this message, then it calculates $C1 - kC2$ and gets point M. Because $C1 - kC2 = M + rk - k(rG) = M + rK - r(kG) = M$, then M is decrypted to get plaintext.

2.2 Improved ECC

Traditional ECC [5, 7, 12, 17, 18] has a big computation burden due to inversion operation. Hence, we improve ECC by ignoring inversion which has a high efficiency.

1) Signer notarizes Hash function to generate information abstract.
2) Signer determines elliptic curve parameter $F = (P, a, h, g, n, h)$ or $(m, f(x), a, h, g, n, h)$.
3) Signer sends determined Hash function and elliptic curve parameter to verifier.
4) Signer chooses key x on the basis of finite field $G(P)$ and selected elliptic curve point group. Then it gets public key $y = xg$ and public y.
5) Signer selects random number $K, 1 \leq K \leq n-1$.
6) It computes $r = kg$, if $r = 0$ then return back step 5.
7) It computes $s = mrx - k$ and gets (s, r) as the signature of m. (s, r) and m are sent to verifier.
8) Verifier calculates $r' = sg + myr$.
9) Verifier judges whether $n' = r$, if they are equal, signature is properly. Otherwise, it rejects signature.

3 Proposed Scheme

3.1 Homomorphic Encryption

The ciphertext can be operated directly without decryption by Homomorphic encryption. Setting encryption function is E_{k1}, decryption function is D_{k2}, plaintext is $M = m_1, m_2, \cdots, m_n$. α and β denote operation. If encryption and decryption function satisfy Homomorphic encryption property, then the following formula is correct.

$$\begin{aligned} & \alpha(E_{k1}(m_1), E_{k2}(m_2), \cdots, E_{kn}(m_n)) \\ = \; & \beta(E_{k1}(m_1, m_2, \cdots, m_n)). \end{aligned} \tag{1}$$

When data $m_1, m_2, \cdots, m_n$ conducts β operation without leaking, we can encrypt it as $(E_{k1}(m_1), E_{k2}(m_2), \cdots, E_{kn}(m_n)$, then do α operation for it. The result is decrypted as $\beta m_1, m_2, \cdots, m_n$. The addition homomorphism and multiplication homomorphism can be expressed as:

$$\begin{aligned} & m_1 + m_2 + \cdots + m_n \\ = \; & D_k(E_k(m_1) + E_k(m_2) + \cdots + E_k(m_n)). \\ & m_1 \cdot m_2 \cdots m_n \\ = \; & D_k(E_k(m_1) \cdot E_k(m_2) \cdot \cdots \cdot E_k(m_n)). \end{aligned}$$

3.2 Improved ECC Homomorphic Encryption

We use the improved ECC to realize the addition homomorphism and multiplication homomorphism.

1) Homomorphic addition.

Plaintext m_i is coded on one point P_{m_i} in E. Randomly select a number r_i and get encrypted data (C_{1_i}, C_{2_i}). It makes additive operation for $(C_{1_l}, C_{2_l}) \cdots (C_{1_n}, C_{2_n})$ and obtains $(\sum_{i=1}^{n} C_{1_i}, \sum_{i=1}^{n} C_{2_i})$. Then calculate $C = k\sum_{i=1}^{n} C_{1_i}$. So we can prove:

$$\begin{aligned} k\sum_{i=1}^{n} C_{1_i} &= kG\sum_{i=1}^{n} r_i = k\sum_{i=1}^{n} r_i. \\ \sum_{i=1}^{n} C_{2_i} - C &= k\sum_{i=1}^{n} r_i + \sum_{i=1}^{n} P_{m_i} - k\sum_{i=1}^{n} r_i \\ &= \sum_{i=1}^{n} P_{m_i}. \end{aligned}$$

So we can get sum $\sum_{i=1}^{n} P_{m_i}$, and decrypt it to obtain $\sum_{i=1}^{n} m_i$.

2) Homomorphic multiplication. Plaintext m_i is calculated, then it gets $(C_{1_i}, C_{2_i}, C_{3_i})$. It makes multiplication operation for $(C_{1_l}, C_{3_l}) \cdots (C_{1_n}, C_{3_n})$ and obtains $(C_{1_l} \cdot C_{2_l} \cdots C_{1_n}, C_{3_l} \cdot C_{3_l} \cdots C_{3_n})$. Then calculate $k^n \cdot C_{1_l} \cdot C_{1_2} \cdot C_{1_n}$ through private key k. So we can prove:

$$\begin{aligned} k^n \cdot C_{1_l} \cdot C_{1_2} \cdot C_{1_n} &= k^n G^n r_1 \cdot r_2 \cdots r_n \\ &= C_{2_l} \cdot C_{2_2} \cdots C_{2_n}. \end{aligned}$$

So we can get $C_{3_l} \cdot C_{3_2} \cdots C_{3_n} \cdot C_{2_l}^{-1} \cdot C_{2_2}^{-1} \cdot C_{2_n}^{-1} = m_1 \cdot m_2 \cdots m_n$.

4 Experiments and Analysis

In order to verify the effectiveness of proposed medical image encryption, we select two medical images as input image conducted on MATLAB. Figures 1, 2 are the original images and histograms. Figures 3, 4 are the encrypted images and histograms. Figures 5, 6 are the decrypted images and histograms.

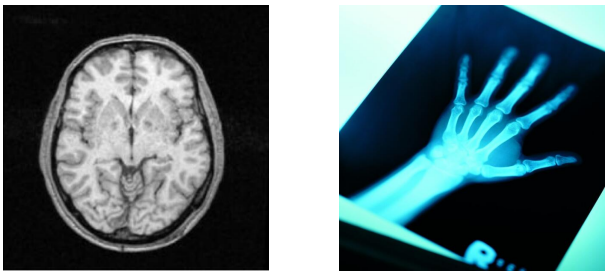


Figure 1: Original images

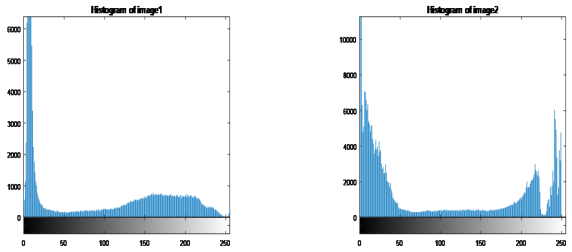


Figure 2: Histogram of original images

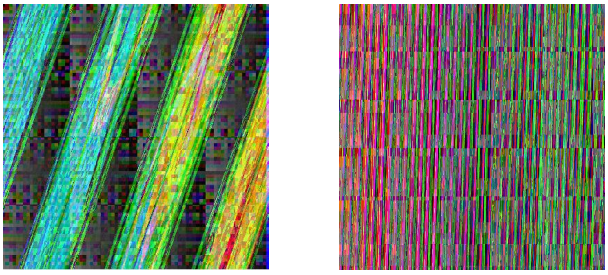


Figure 3: Encrypted images

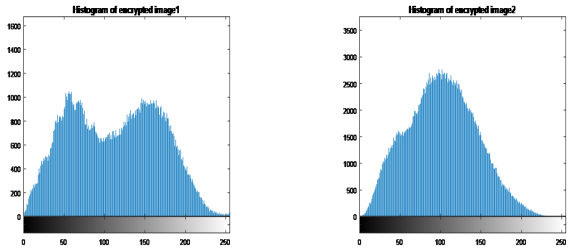


Figure 4: Histogram of encrypted images

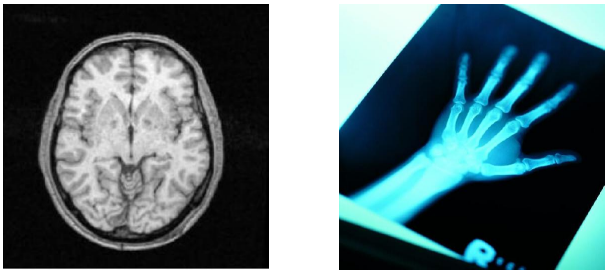


Figure 5: Decrypted images

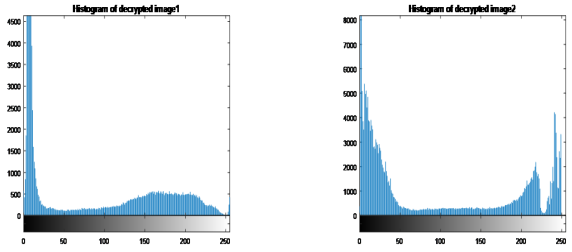


Figure 6: Histogram of decrypted images

Table 1: Correlation comparison between adjacent pixels

Correlation	vertical direction	Horizontal direction	Diagonal direction
Image1	0.9241	0.9235	0.9417
Encrypted image1	0.0015	0.0008	0.0021
Image2	0.9221	0.8719	0.9426
Encrypted image2	0.0041	0.0022	0.0018

4.1 Key Space Analysis

We adopt improved ECC to encrypt image, which has eight keys. If the computer accurates to 10^{-15}, the space size of the key is 10^{128}. The key space is large enough to resist the exhaustive attack.

4.2 Sensitivity Analysis

It is sensitive to system parameters and initial values, which means that if the initial value changes slightly, the decrypted image will not be associated with the original image. As shown in Figure 3, during the decryption process, key adds 0.1^8 to decrypt medical image. Based on the above theory, the algorithm is sensitive to key, which indicates that it has the ability to resist the exhaustive attack.

4.3 Correlation Analysis of Adjacent Pixels

We use the following formulas to calculate the correlation coefficients.

$$\begin{aligned} E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i. \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N}[x_i - E(x)]^2. \\ Cov(x,y) &= \frac{1}{N}\sum_{i=1}^{N}[x_i - E(x)][y_i - E(y)]. \\ g_{xy} &= \frac{Cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}. \end{aligned}$$

Where x and y denote two adjacent pixel values in the image and g_{xy} is correlation coefficient between adjacent pixels shown in Table 1.

4.4 Information Entropy

Information entropy denotes the degree of uncertainty system, and it is used to describe the uncertainty of image information. The information entropy can be used to analyze the distribution of gray value in the image. Let $P(m_i)$ be proportion of pixel with gray value m_i in image and $\sum_{i=0}^{255} P(m_i) = 1$. The information entropy of the pixel is defined as:

$$H(m) = -\sum_{i=0}^{255} P(m_i \log_2 P(m_i)).$$

We make comparison with HHC [12], CST [3] and CTM [9] as shown in Table 2.

Table 2: Information entropy comparison

Method	Encrypted image1	Encrypted image2
HHC	0.712	0.708
CST	0.687	0.693
CTM	0.667	0.689
Proposed	0.796	0.797

4.5 Plaintext Sensitivity Analysis

Differential attack: A small change in the original image can cause a huge change in the encrypted image. The attacker can obtain the connection between the original image and the encrypted image. We adopt number of pixel change rate (NPCR) and unified average changing intensity (UACI) to measure it. They are defined as:

$$\begin{aligned} NPCR &= \sum_{ij} D(i,j)/m \times n. \\ UACI &= \frac{1}{m \times n}[\sum_{ij} \frac{|C_1(i,j) - C_2(i,j)|}{255}]. \end{aligned}$$

Where m and n represent the row and column of the image respectively. C1 and C2 are obtained by changing only one pixel value of the original image. $C1(i,j)$ and $C2(i,j)$ represent the pixel values in the (i,j) coordinate.

NPCR and UACI values are shown in Table 3 and 4, the tiny change in the original image can make the encryption image close to 100% of NPCR changes, the encrypted image's average change is above 30% (UACI). At the same time, it also shows that image information spreads to the cipher image well, compared with the HHC, CST and CTM, the proposed algorithm has very good sensitivity, robustness for the differential attack.

Table 3: Encrypted image1

Method	NPCR%	UACI%
HHC	89.67	31.02
CST	91.42	37.62
CTM	92.14	38.54
Proposed	99.23	39.58

Table 4: Encrypted image2

Method	NPCR%	UACI%
HHC	90.76	32.01
CST	91.12	35.53
CTM	91.47	36.45
Proposed	99.18	38.59

5 Conclusion

In this paper, a new medical image encryption algorithm is proposed based on improved elliptic curve cryptography by combining with homomorphic encryption. We analyze the districts of traditional ECC, then we modify it. The experimental results show that the algorithm has better key space with better encryption effect and higher key sensitivity. In addition, the algorithm has strong robustness for resisting statistical attack and exhaustive attack. In the future, in terms of medical image encryption, we will adopt some deep learning models to study it.

References

[1] S. E. Assad, M. Farajallah, "A new chaos-based image encryption system," *Signal Processing Image Communication*, vol. 41, pp. 144-157, 2016.

[2] S. Farwa, T. Shah, N. Muhammad, *et al.* "An image encryption technique based on chaotic s-box and srnold transform," *International Journal of Advanced Computer Science & Applications*, vol. 8, no. 6, 2017.

[3] M. Ghebleh, A. Kanso, "A novel efficient image encryption scheme based on chained skew tent maps," *Neural Computing & Applications*, vol. 4, pp. 1-16 2017.

[4] E. Hariyanto, R. Rahim, "Arnold's cat map algorithm in digital image encryption," *International Journal of Science & Research*, vol. 5, no. 10, pp. 6-391, 2016.

[5] M. S. Hwang, C. C. Lee, J. Z. Lee, and C. C. Yang, "A secure protocol for bluetooth piconets using elliptic curve cryptography," *Telecommunication Systems*, vol. 29, no. 3, pp. 165-180, 2005.

[6] M. S. Hwang and I. C. Lin, *Introduction to Information and Network Security (6ed, in Chinese)*, Taiwan: Mc Graw Hill, 2017.

[7] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves," *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 73-84, Mar. 2004.

[8] S. Khatoon, T. Thakur, B. Singh, "A provable secure and escrow-able authenticated group key agreement protocol without NAXOS trick," *International Journal of Computer Applications*, vol. 171, no. 3, pp. 1-8, 2017.

[9] C. Li, G. Luo, K. Qin, *et al.* "An image encryption scheme based on chaotic tent map," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 127-133, 2017.

[10] J. Liu, S. L. Yin, H. Li and L. Teng, "A density-based clustering method for k-anonymity privacy protection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 1, pp. 12-18, Jan. 2017.

[11] S. Mccarthy, N. Smyth, E. O'Sullivan, "A practical implementation of identity-based encryption over NTRU lattices," *IMA International Conference on Cryptography and Coding*, pp. 227-246, 2017.

[12] A. Y. Niyat, M. H. Moattar, M. N. Torshiz, "Color image encryption based on hybrid hyper-chaotic system and cellular automata," *Optics & Lasers in Engineering*, vol. 90, pp. 225-237, 2017.

[13] S. Rajendran, M. Doraipandian, "Chaotic map based random image steganography using LSB technique," *International Journal of Network Security*, vol. 19, no. 4, pp. 593-598, 2017.

[14] H. Sharma, N. Khatri, "An image encryption scheme using chaotic sequence for pixel scrambling and DFrFT," *Proceedings of First International Conference on Smart System, Innovations and Computing*, pp. 487-493, 2018.

[15] X. Su, W. Li, H. Hu, "Cryptanalysis of a chaos-based image encryption scheme combining DNA coding and entropy," *Multimedia Tools & Applications*, vol. 76, no. 12, pp. 1-13, 2016.

[16] L. Teng, H. Li, S. Yin, "A multi-keyword search algorithm based on polynomial function and safety inner-product method in secure cloud environment," *International Journal of Network Security*, vol. 8, no. 2, pp. 413-422, 2017.

[17] L. Teng, H. Li, J. Liu and S. Yin, "An efficient and secure cipher-text retrieval scheme based on mixed homomorphic encryption and multi-attribute sorting method," *International Journal of Network Security*, vol. 20, no. 5, pp. 872-878, 2018.

[18] 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.

[19] S. L. Yin and J. Liu, "A k-means approach for map-reduce model and social network privacy protection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 6, pp. 1215-1221, Nov. 2016.

[20] S. Yin, L. Teng, J. Liu, "Distributed searchable asymmetric encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 4, no. 3, pp. 684-694, 2016.

[21] Q. Zhang, L. T. Yang, X. Liu, Z. Chen, and P. Li, "A tucker deep computation model for mobile multimedia feature learning," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 13, no. 3, pp. 1-39:18, 2017.

Biography

Shoulin Yin biography. He received the B.Eng. and M.Eng. degree from Shenyang Normal University, Shenyang, Liaoning province, China in 2016 and 2013 respectively. Now, he is a doctor in Harbin Institute of Technology. His research interests include Multimedia Security, Network Security, Filter Algorithm, image processing and Data Mining. Email:352720214@qq.com.

Jie Liu biography. Jie Liu is a full professor in Software College, Shenyang Normal University. He received his B.S. and M.S. degrees from Harbin Institute of Technology. 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. Professor Liu had published more than 30 international journal papers (SCI or EI journals) on the above research fields. Email:ljnan127@163.com.

Lin Teng biography. She received the B.Eng. degree from Shenyang Normal University, Shenyang , Liaoning province, China in 2016. Now, she is a laboratory assistant in Software College, Shenyang Normal University. Her research interests include Multimedia Security, Network Security, Filter Algorithm and Data Mining. Email:910675024@qq.com.

A Multi-threading Solution to Multimedia Traffic in NIDS Based on Hybrid Genetic Algorithm

Xu Zhao[1], Guangqiu Huang[2], and Reza Mousoli[3]
(Corresponding author: Xu Zhao)
School of Computer Science, Xi'an Polytechnic University[1]
No. 19, Jinhua South Road, 710048, Xi'an, China
(Email: 37274679@qq.com)
School of Management, Xi'an University of Architecture and Technology[2]
No. 13, Yanta Road, Beilin District, 710055, Xi'an, China
School of Law, Criminal Justice and Computing, Canterbury Christ Church University[3]
North Holmes Road, CT1 1QU, Canterbury, UK
(Received July 27, 2018; Revised and Accepted Jan. 24, 2019; First Online July 30, 2019)

Abstract

Packet omission and subsequently data loss is inevitable when the network traffic exceeds the load capacity threshold of Network Intrusion Detection System (NIDS). In these circumstances, relatively dangerous packets should be given priority for processing by NIDS. To address this problem and offer a possible remedy, this paper proposes a multi-threading solution specifically for multimedia packets in NIDS by using two different genetic algorithm. In this solution, two optimization objectives are achieved simultaneously: One is to maximize the sum of danger coefficient of multimedia packets in every thread and the other is to process the workload of each thread at its maximum workload. This paper also compares the advantages of hybrid genetic algorithm to simple genetic algorithm in the implementation process of the proposed solutions. By using this proposed solution, NIDS can identify multimedia packets and then select the more dangerous multimedia packets for processing within the maximum processing capacity of different threads when packet omission occurs. Experimental results indicate that this solution can help NIDS to improve its differentiation and selection ability for dangerous multimedia packets effectively.

Keywords: Danger Coefficient; Genetic Algorithm; Multimedia Packets; Network Intrusion Detection System (NIDS); The Solution of Choosing Danger

1 Introduction

1.1 Background

NIDS is a system that detects malicious activities by monitoring network traffic. With the rapid increase of network speed, the requirements of the NIDS's processing efficiency has also increased. Nevertheless, packet omission is inevitable when the network traffic exceeds the load capacity of NIDS. This problem has stimulated extensive research studies on how to reduce the omission ratio of NIDS and how to minimize the security risks when the omission becomes inevitable. Among these reseach activities for finding a optimal solution, artificial intelligence has attracted considerable interest from the research community and various artificial intelligence algorithms in improving NIDS has been investigated, such as Artificial Neural Networks [9, 12], Clustering algorithm [3, 7, 11], Particle Swarm Optimization [1, 4, 10] and genetic algorithm [2, 5, 13, 14]. However, artificial intelligence algorithms have not been found for multimedia traffic analysis for NIDS, and the study in this paper addresses this gap in order to solve this shortcoming.

With the rapid development of high-speed networks, the proportion of multimedia packets in the network traffic is increasing. The targeted method of processing multimedia packets can greatly improve the efficiency of NIDS. Previously we have proposed an identifying method and two separate processing methods for multimedia packets to raise the efficiency of the NIDS and have gained satisfactory results [6].

Because of the various types of multimedia, the security of multimedia packets can vary according to different types of packets. Under the premise of limited system processing capacity, the more dangerous multimedia packets should be given priority for processing when the network traffic is too great and packet omission becomes inevitable. On this basis, we propose a multi-threading solution to multimedia packets in NIDS systems based on genetic algorithm. When packet omission occurs, this solution can select more dangerous multimedia packets for processing within the maximum processing capacity of different threads.

1.2 Contributions Summary

The main contributions of this paper are:

1) The study of multimedia packets relating to NIDS in the networks. On the basis of previous studies [6, 8, 15, 16] for multimedia packets, we propose a multi-threading solution for multimedia packets based on hybrid genetic algorithm. By using this solution, NIDS can focus its limited processing power on more dangerous multimedia packets when omission becomes inevitable.

2) We propose two optimization objectives to optimize NIDS:

   - The sum of danger coefficients of multimedia packets in every threads is maximal.
   - When the above objective is achieved, the load of each thread exacctly reaches the highest.

3) We introduce two concrete implementations with simply genetic algorithm and hybrid genetic algorithm of the solution. The advantages and disadvantages of the two solutions are also analyzed. In addition, we also design several experiments which compare the packet loss rate, the sum of danger coefficients, the detection number of multimedia packets and the detection rate of dangerous incident when using different solutions. We also prove its effectiveness based on above-mentioned experimental results.

1.3 Paper Organization

The rest of the paper is organized as follows: First, related studies are discussed in Section 2; Section 3 presents the description of the solution; Section 4 describes the determination of the value of parameters. Section 5 presents the implementation of the solution with two different genetic algorithms; The experiment and an analysis of the result for contrasting the differences before and after using the solution is in Section 6; Section 7 summarizes the whole paper and presents some directions for the future work.

2 Related Work

2.1 Artificial Intelligent Algorithms on NIDS

Artificial intelligent algorithms usually offer an automatic mechanism to enhance the performance of NIDS. Here are several common algorithms.

1) Artificial Neural Networks [3,9,12]: DeLima [12] presented a method for building a prototype for NIDS, which uses an artificial neural network as a detection mechanism. Nevertheless, the adjustment of weights is somewhat complex.

2) Clustering algorithm [1,4,7,11]: Chandrashekhar [11] proposed an efficient intrusion detection model by amalgamating competent data mining techniques such as K-means clustering, Multilayer layer perception (MLP) neural network and support vector machine (SVM).This model can improve the prediction of network intrusions, but it also has problems. For example, it might mistake dubious data for normal data.

3) Particle Swarm Optimization [2,5,10,13]: Cleetus [5] proposed an intrusion detection solution based on particle swarm optimization by using multiobjective functions. This solution has a strong global search capability which is used for dimensionality optimization. However, it is easy to fall into local optimum.

4) Genetic algorithm [6, 8, 14, 15]: In the field of artificial intelligence, genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions.

Samaneh Rastegari [14] proposed a solution that uses genetic algorithm to evolve a set of simple, interval-based rules based on statistical, continuous-valued input data. This new approach provides a very compact set of simple, human-readable rules with strongly competitive detection performance in comparison to other machine learning techniques. But this approach should be modified for multi-class classification, to discover rulesets that can identify which kind of attack is taking place.

Additionally, although [6] using GA to improve the network attack detection accuracy, but without considering the GA individual selection adjustment, and it only uses KDD dataset to validate their methods, and not for practical applications.

Furthermore, Dustin Y. Harvey [8] described a method using GA to identify irregular network intrusion. The process includes both quantitative and definite features of network data for deriving classification rules. Though, the addition of quantitative feature can amplify the detection rate no tentative results are present.

2.2 The Identifying and Processing Methods for Multimedia Packets

With the increasing speed of network, the proportion of multimedia packets in network traffic is increasing. Compared to other packets, because the multimedia packets are relatively safe, the NIDS has less detection rules for specific multimedia types [16]. Therefore, the recognition of multimedia packets and the separate processing according to different multimedia types will greatly improve the performance of NIDS. O.Marques of Florida Atlantic University originally proposed this idea, but he did not give specific solutions [15]. We have put forward a series of

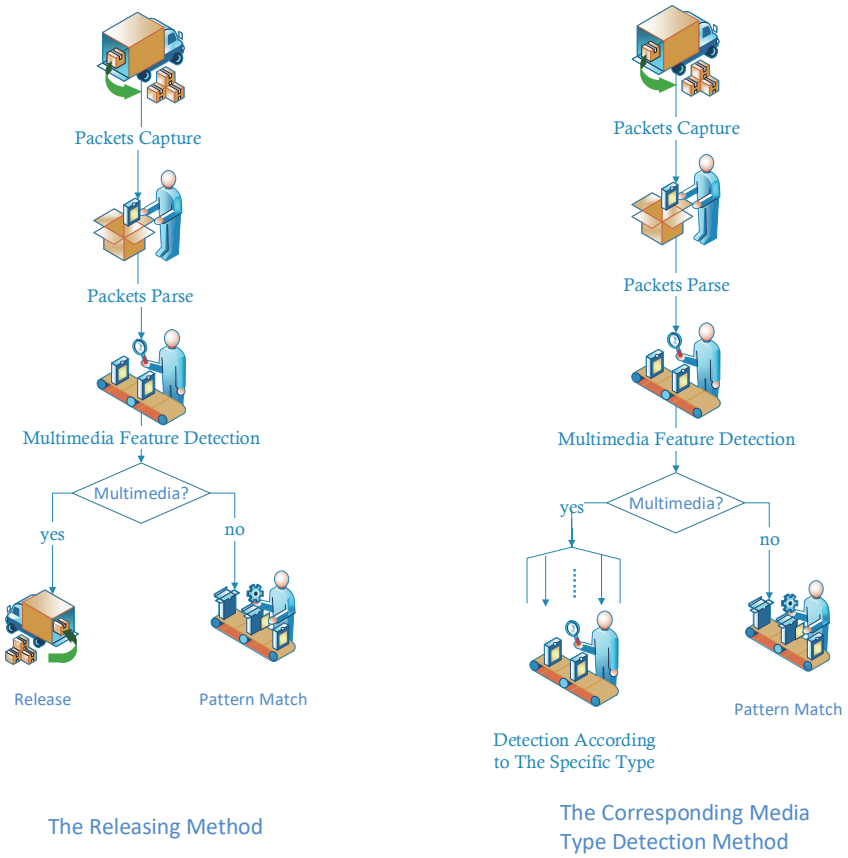


Figure 1: The work flow of two methods *** Please redraw this figure. Part of the icons (pictures) maybe have copyright problem. (Please don't use the commercial picture or icons). Please use a "rectangle with text" to replace the icons. ***

processing methods [17–20] of NIDS for multimedia packets with good results. In the meantime, O. Marques and Pierre have also carried out follow-up studies [21], but their focus is mainly on the loopholes of streaming and non streaming specific multimedia files. Similarly, Zander of the Murdoch University in Australia proposed the classification [22] of multimedia traffic in the firewall by machine learning technology, which provides a reference for the depth detection method of multimedia packets.

Specifically, we have proposed an identifying method and two particular processing methods for multimedia packets in [16]. The principle of the identifying method for multimedia packets is to detect the multimedia features information at the front of multimedia packets.

The two processing methods are the Releasing method and the CMTD (corresponding media type detection) method. Figure 1 illustrates the work flow of two methods. The Releasing method lets identified the multimedia packet skip over the conventional detection process. Though this method is simple and efficient, its security is lower.

The CMTD method [16] is a safe and efficient method. In order to achieve it, a multimedia rule base is created. The multimedia rule base stores the rules which are collected specifically for multimedia packets. The CMTD method can be used to choose the corresponding multimedia rules according to the specific multimedia type that the packet carries in order to pre-detect intrusive characteristics. If there is no problem, it is released immediately; if there exists a problem, put it into the conventional detection process. Because there are far fewer multimedia rules than rules for conventional detection process in NIDS, this method can significantly improve the detection efficiency for most of the safe multimedia packets. The safety of this method is also greater than that of the releasing method.

Although these methods can raise the efficiency of NIDS, they are mainly suitable for no-omission. On this basis, we propose a multithreading solution to multimedia packets in NIDS based on genetic algorithm. When omission occurs this solution can choose more dangerous multimedia packets for processing within the maximum processing capacity of different threads.

3 Description of Solution

According to the MIME protocol, the multimedia packet types in the network are as many as 133. Nevertheless, the risk of every type of multimedia packets is different. For example, octet-stream*.exe is more dangerous than others. According to the set-method of the danger co-

efficient of multimedia types in [20], the multithreading solution to multimedia packets in NIDS can be described as following:

N multimedia packets $P_1, P_2, \cdots, P_n$ have been captured, let the load of each thread in NIDS be set as LT, then the load of these multimedia packets to NIDS is $L(P_i) \in (0, LT](i = 1, 2, \ldots, n)$ and the danger coefficient of these multimedia packets is $D_k(P_i)(k = 1, 2, \ldots, 133, 1 \leq i \leq n)$.The key point is how to determine the distribution solution which can make the highest sum of danger coefficient of multimedia packets in each thread and guarantee the load of these packets is less than the load of each thread.

4 How to Determine the Value of Parameters $L(P_i)$ and $D(P_i)$

$L(P_i)$: The load to the system which is caused by multimedia packet Pi. Because there are significant positive correlations [17] between the time complexity of the pattern matching algorithm and length of the string to be matched, $L(P_i)$ is determined by the ratio of the actual length of the packet load and the total length.

$D(P_i)$: The value of $D(P_i)$ should be set for different media types according to the degree of risky information [17] carried by packets (as shown in Table 1). For instance, executable files can appear in multimedia files of octet-stream type, the value of D(Pi) of octet-stream type can be set higher. The table below shows the value of $D(P_i)$ for several common multimedia types.

Table 1: The value of P for several common multimedia types

multimedia types	file type	$D(P_i)$
octet-stream	exe rar	3.0
x-JavaScript	js	2.1
x-tar	tar	2.6
jpeg	Jpzjpg jpeg	1.5
gif	gif	1.5
html	htm html hts	1.3
x-shockwave-flash	swf swfi	1.8
.....	.....	.....

5 Implementation of the Solution

To solve the above-mentioned problem, we will illustrate the solution by two genetic algorithms in the following sections and compare the differences between them. One is simple genetic algorithm, the other is Hybrid genetic algorithm which consists of the simple genetic algorithm and the FFD approximation algorithm.

5.1 The Solution by Using Simple Genetic Algorithm

Step 1: Individuals Code. Because the operation object of genetic algorithm is the symbol string which indicates individual, the operation object must be encoded as the symbol string in this solution. The chromosome coding method is defined as follows:

Chromosome coding method: Let K thread number be $T_1, T_2, T_3, \ldots\ldots T_k, (k \leq n)$, n multimedia packets will be loaded into these K threads. The number sequence of each multimedia packet $P_i(i = 1, 2, ..., n)$ which is loaded into these threads constitutes the chromosome coding of this problem. For example, $T_1T_3T_1T_2...T_2T_1$ means that multimedia packets P_1, P_2, P_3 are loaded into the thread T_1, multimedia package P_2 is loaded into thread T_3, *etc.* The initial population can be generated by random permutation of $T_1, T_2, T_3, \ldots\ldots T_k$.

Step 2: Initialization. Let the evolution generations counter be t and give t an initial value 0. Let the maximum value be T; the initial population P(0) can be generated by random permutation as in Step 1.

Step 3: Evaluation of the fitness value: P(t) indicates the population that evolves to the t generation. Genetic Algorithm determines the probability of an individual in the current population P(t) to the next generation population by adapting the proportional probability of individual fitness. In order to estimate this probability correctly, every individual fitness in population P(t)must be calculated.

The objective function and the fitness function: Let m be the number of threads used in a distribution scheme, and let $T(P_i)$ be the number of the thread in which multimedia packet P_i is loaded. Besides, let S_j be the sum of load of the multimedia packets in thread T_j. In order to make the best use of all threads, the optimization objective function can be written as Equation (1).

$$\begin{aligned} f(x) &= m \cdot \{m - \sum_{j=1}^{m} s_j\} \\ &= m \cdot \{m - \sum_{j=1}^{m} [\sum_{T(P_i)=T_j} L(P_i) \\ &\quad - a \cdot \max(0, \sum_{T(P_i)=T_j} L(P_i) - 1)]\}. \end{aligned} \tag{1}$$

The danger coefficient objective function of multimedia packets in each thread is:

$$max \sum P_i D_i, 1 \leq i \leq n \tag{2}$$

In Equation (1), a indicates the penalty factor when the sum of load of the multimedia packets in thread T_j exceeds the load of thread T_j. These two objective

functions not only maximize the sum of danger coefficients of multimedia packets in every thread, but also make each thread reach the highest load. The fitness function is:

$$F(X) = \begin{cases} C_{\max} - f(X), f(X) < C_{\max} \\ 0, f(X) \geq C_{\max} \end{cases} \quad (3)$$

In Equation (3), $C_{\max}$ indicates an appropriate positive which adjusts the fitness function to take a non-negative value.

Step 4: Selection operation. The selection operator can adopt the proportional selection operator. $P^{'}(t)$ can be obtained when the selection operator acts on the population $P^{'}(t)$.

Step 5: Crossover operation. The crossover operator can use the single point crossover operator. $P^{''}(t)$ can be obtained when the crossover operator acts on the population $P^{'}(t)$.

Step 6: Mutation operation. The mutation operator can adopt uniform random variation in the coded character set $V = \{T_1, T_2, T_3, \ldots\ldots T_k\}$. When the mutation operator acts on the population $P^{''}(t)$, $P^{'''}(t+1)$ will be achieved after selection, crossover and mutation operation of $P^{''}(t)$.

Step 7: The judgment of termination condition. When $t <= T$, then $t \leftarrow t+1$. As a new population, $P^{'''}(t+1)$ will replace $P(t+1)$.Next go to Step 2 and start the next cycle. If $t > T$, then the population with the greatest fitness in evolutionary process will be output as the optimal solution and computation will terminate.

5.2 The Disadvantage of the Above mentioned Solution

The disadvantage of the abovementioned solution is that some invalid chromosomes would be generated in initialization and evolutionary process. In the distribution scheme represented by these invalid chromosomes, the sum of load of the multimedia packets in a thread will exceed the load of this thread. This will reduce the operating efficiency of NIDS.

In view of the above-mentioned facts, we propose the following solution by using a hybrid genetic algorithm which consists of the simple genetic algorithm and the FFD approximation algorithm.

5.3 The Solution Combined with FFD Approximation Algorithm

According to the principle of the FFD approximation algorithm, all multimedia packets are firstly listed in descending order according to the amount of load for each packet. Then these packets are distributed to the threads.

The above approaches are applied to the decoding process of the chromosome chain (Step 1 above) in genetic algorithm. Their specific steps are described below:

1) All multimedia packets are listed in descending order according to the amount of load for each packet.

2) The above-mentioned packets are distributed to all threads. When the sum of load of the multimedia packets in thread T_j exceeds the load of thread T_j, then the extra multimedia packets will be distributed to thread T_{j+1}.

In the case of the sum of load of the multimedia packets in thread T_m exceeds the load of thread T_m, the extra multimedia packets will be distributed to a new thread, and $m \leftarrow m+1$.

5.4 The Advantage of the Improved Solution

By using the above method, not only does the distribution scheme of the above steps meet the principle of FFD approximation algorithm, but also the sum of load of the multimedia packets in each thread does not exceed the load of thread. So, Penalty function is not required when the objective function is calculated. The objective function is listed in Equation (4).

$$\begin{aligned} f(x) &= m \cdot \left\{ m - \sum_{j=1}^{m} s_j \right\} \\ &= m \cdot \left\{ m - \sum_{j=1}^{m} \sum_{T(P_i)=T_j} L(P_i) \right\} \end{aligned} \quad (4)$$

The Fitness value can also be the value of the above objective function, which is described as following:

$$F(X) = f(X). \quad (5)$$

6 Experiment and Result Analysis

6.1 Experimental Environment

The experiments reported here demonstrate a variety of changes before and after using the multithreading solution to multimedia packets. Experimental environment consists of three computers which are configured as follows:

CPU: Intel Core i7 5960X (16 threads);

Memory: 8 GB DDR4;

OS: ubuntu-18.04.1.

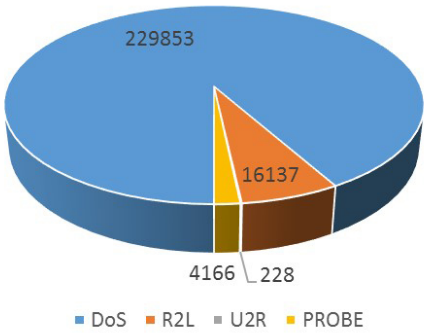


Figure 2: The types and quantities of attack category in attack traffic

Experimental data is a mixture of MIT Lincoln Laboratory KDD CUP 99 data sets and the background traffic. KDD CUP 99 data sets include four types of network attacks [22], DoS, R2L, U2R and PROBE. The types and quantities of attack category are shown as follows:

Background traffic is the real flow captured in the network, containing a large number of multimedia packets. In the experiment, the background traffic is sent by the first computer. As the attacker, the second computer uses Lincoln Laboratory KDD CUP 99 data set and IDS Informer to generate attack traffic. Both mixed traffic are sent to test NIDS installed on the third computer, as is shown in Figure 3.

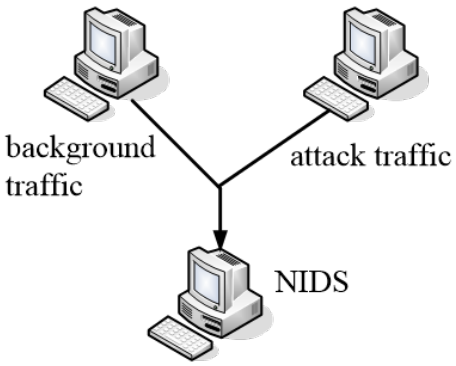


Figure 3: Experimental environment *** Please redraw this figure. Part of the icons (pictures) maybe have copyright problem. (Please don't use the commercial picture or icons). Please use a "rectangle with text" to replace the icons. ***

In order to complete each test successfully, we add the program to the preprocessor (spp_stream4 file) of NIDS for record the danger coefficient of multimedia packets which is selected into different threads. In addition, we improve the transmission speed of the mixed traffic so as to obtain experimental results in the case of packet loss.

All kinds of multimedia packet information in background traffic are shown in **Table 2.**

Figure 4-6 are analyses of the background traffic. As can be seen from the figure, the number of documents such as ASP and ASPX is the largest, exceeding 1200. On the total amount of data, the sum of JS files is the largest, reaching 3MKB. On the average detection length, the SWF file is the longest, with an average of more than 200KB.

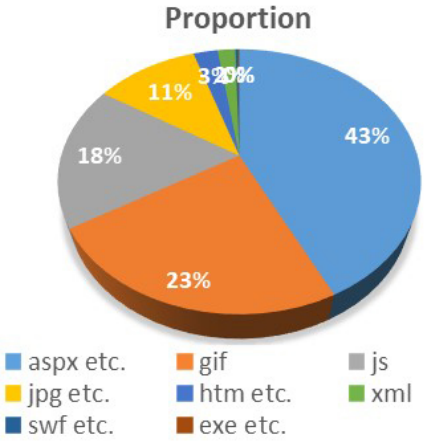


Figure 4: The proportion of different types of multimedia files

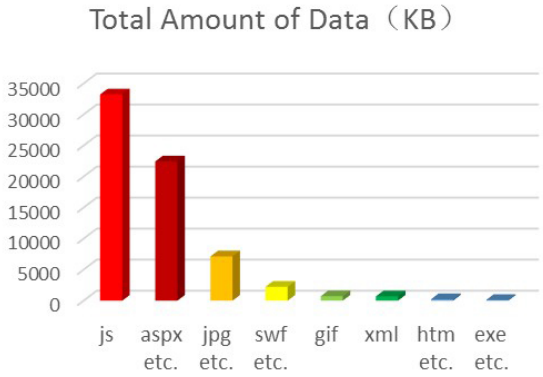


Figure 5: The total amount of data of different types of multimedia files

6.2 The Sum of Danger Coefficient in Different Threads

The first experiment is to compare differences in the sum of danger coefficients of the multimedia packets which are selected into each thread at the same time slice when using different methods.

As can be seen from Figure 7, the sum of danger coefficients of the multimedia packets in each thread has obviously increased after using the simple GA-based multithreading solution. The reason for this improvement is that these multimedia packets are selected randomly into each thread and its danger coefficient is not a consideration before using the GA-based solution. Therefore, the sum of danger coefficients is low and random. Nevertheless this value is relatively higher and stable after using the GA-based solution. This can also be shown by the sample variance of results. According to the following

Table 2: All kinds of multimedia packet information in background traffic

MIME Type	File Type	number	Total Amount of Data(KB)	Average Length(KB)	Risk factor
application/octet-stream	exe bin rar etc.	3	121	40	3.0
x-javascript	Js	545	33245	61	2.5
text/html	htm html hts etc.	81	243	3	1.6
application/x-asap etc.	asp aspx jsp etc.	1320	22440	17	1.6
text/xml application/xml	xml	59	708	12	1.3
image/jpeg	Jpz jpg jpeg	340	7140	21	1.5
image/gif	gif	728	728	1	1.5
x-shockwave-flash	swf swfi	10	2250	225	1.8

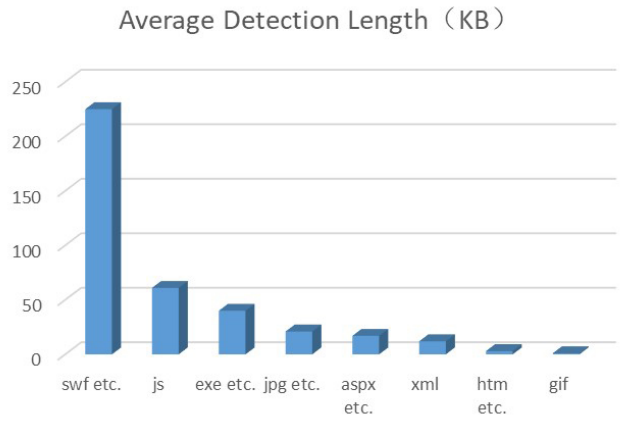


Figure 6: Average detection length for different types of multimedia files

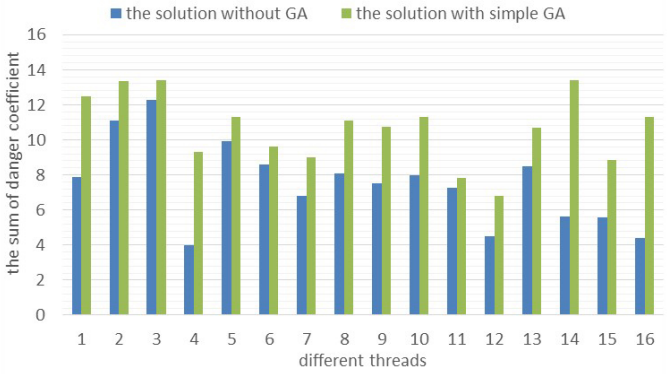


Figure 7: The differences in the sum of danger coefficient between using GA and not using GA

Figure 8: The differences in the sum of danger coefficient between using simple GA and hybrid GA

calculation formula of sample variance, sample variance is 6.5 before improvement, while it is 4.2 after improvement.

$$S^2 = \frac{\sum_{i=1}^{n} (x_i - E(x))^2}{n-1} \quad (6)$$

Figure 8 shows the differences in the sum of danger coefficient between using simple GA and hybrid GA. As you can see in the Figure 8, their difference is not obvious. In 69% of the threads, the sum of danger coefficient caused by using hybrid GA is more than that caused by simple GA.

6.3 The Packet Loss Rate

The second experiment is to compare the packet loss rate in three different situations.

As can be seen from Figure 9, after using simple GA or hybrid GA solutions, the packet loss rate increases slightly compared with that before improvement. The main reason for this problem is the high time complexity of GA. In addition, it is also found that the packet loss rate of the hybrid GA solution is lower than that of the simple GA solution. The main reason is that some invalid chromosomes would be generated in initialization and evolutionary process of the simple GA solution. In the distribution scheme represented by these invalid chromosomes, the sum of load of the multimedia packets in a thread will exceed the load of this thread. This leads to lower operation efficiency and increased packet loss rate of NIDS. Nevertheless, the advantage is that Solutions using hybrid GA can maintain packet loss rates within 6% higher than those without GA.

6.4 The Detection Number of Different Types of Multimedia Packets

The third experiment is to compare differences between the detection numbers of various types of multimedia packets with three different solution in the case of packet

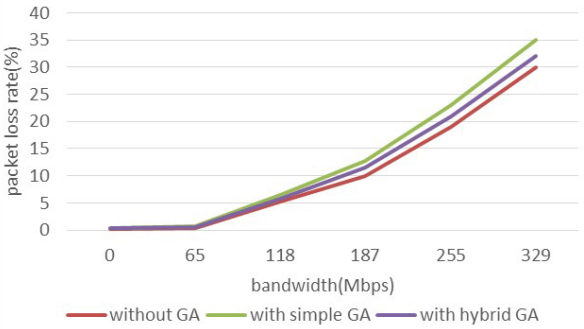


Figure 9: The packet loss rate caused by three different solutions

loss.

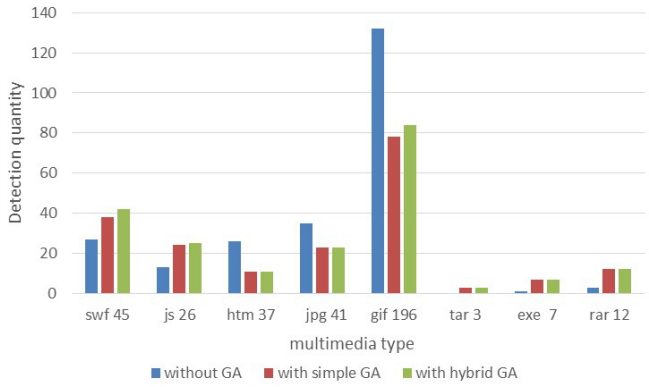


Figure 10: The detection number of different types of multimedia packets

As shown in Figure 10, when using the solution without GA, because all multimedia packets have been selected according to the time sequence they are captured, for all multimedia types, the more the numbers, the more the detection number, such as gif. While after using the solution with simple GA, because NIDS select multimedia packets according to its danger coefficient, the more danger, the more the detection number. For example, the detection number of exe type has increased by 6 times, all exe file are detected. On the other hand, the detection number of multimedia packets with lower danger coefficient has decreased. For instance, the detection number of gif type decreases by 41%. If the solution with hybrid GA is adopted, this trend of improvement will continue. Because the hybrid GA solution is superior to simple GA in controlling thread load capacity, the detection number of multimedia types with large number is slightly higher than that of simple GA.

6.5 The Detection Rate of Dangerous Incident

The fourth experiment is used to test the detection rate for dangerous incident by using three different solutions. The types and numbers of attack flow as shown in Table 3.

These experiments show that, when the network traffic exceeds the processing capacity of NIDS, the limited processing power of NIDS can be focused on the more dangerous multimedia packets by using the multithreading solution.

Table 3: The types and numbers of attack flow

Attack Type	DoS	R2L	U2R	PROBE	Total
Numbers	229853	16137	228	4166	250384

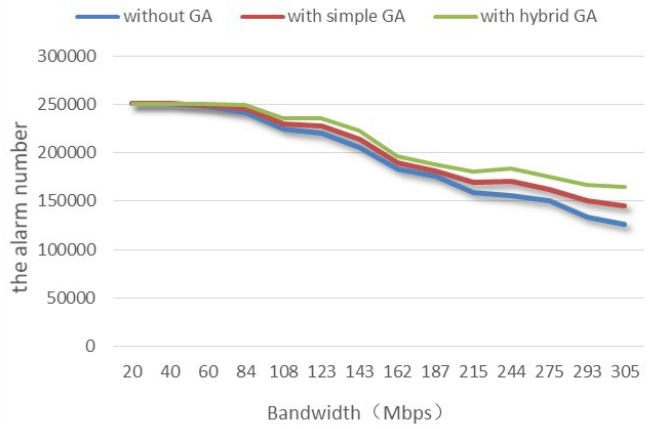


Figure 11: The detection rate using three different solutions

7 Conclusion and Future Work

With the high development and the wide application of network technology, network invasion is becoming an increasingly serious problem for network engineers and managers. Intrusion detection becomes a critical component of network security administration.

This paper addresses the performance challenges of NIDS in high-speed networks by proposing the multithreading solution based on genetic algorithm. By using this solution, when the network traffic is too great and omission is inevitable, NIDS can choose more dangerous multimedia packets for processing within the maximum processing capacity of different threads. Various experiments have shown that the solution can effectively improve the detection number of dangerous multimedia packets.

As for future work, firstly, I will apply this solution to non-multimedia files, such as bat type. Secondly, I will continue to compare the advantages and disadvantage of the solution proposed in the paper with those of other experiments to achieve more objective evaluations in the higher speed network environment.

Acknowledgment

This work is supported by Shaanxi Science and Technology Project (2019KRM153), Xi'an Science and Technol-

ogy Bureau (201805030YD8CG14(8)), Xi'an Beilin District Science and Technology Bureau (GX1708), Shaanxi Education Science Project(SGH18H089).

References

[1] Amrita, K. K. Ravulakollu, "A hybrid intrusion detection system: Integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers, *International Journal of Network Security*, vol. 20, no. 1, pp. 41-55, 2018.

[2] A. M. V. Bharathy, A. M. Basha, "A multi-class classification MCLP model with particle swarm optimization for network intrusion detection," *S?dhan?*, vol. 42, no. 5, pp. 631–640, 2017.

[3] E. M. Boujnouni, 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.

[4] A. M. Chandrashekhar, K. Raghuveer, "Amalgamation of K-means clustering algorithm with standard MLP and SVM based neural networks to implement network intrusion detection system," *Parasitology*, vol. 114, no. 2, pp. 159-73, 2014.

[5] N. Cleetus, K. A. Dhanya, "Multi-objective particle swarm optimization in intrusion detection," *Procedia Computer Science*, vol. 60, no. 1, pp. 714-721, Mar. 2015.

[6] Y. Danane, T. Parvat, "Intrusion detection system using fuzzy genetic algorithm," in *International Conference on Pervasive Computing (ICPC'15)*, 2015. ISBN: 978-1-4799-6272-3.

[7] M. Ghaffari, N. Ghadiri, "Ambiguity-driven fuzzy C-means clustering: How to detect uncertain clustered records," *Applied Soft Computing*, pp. 1-12, 2016.

[8] D. Y. Harvey, M. D. Todd, "Automated feature design for numeric sequence classification by genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 1, 2014.

[9] E. Hodo, X. Bellekens, A. Hamilton, *et al.*, "Threat analysis of IoT networks using artificial neural network intrusion detection system," *Tetrahedron Letters*, vol. 42, no. 39, pp. 6865-6867, 2017.

[10] R. Kondaiah, B. Sathyanarayana, "Trust factor and fuzzy firefly integrated particle swarm optimization based intrusion detection and prevention system for secure routing of MANET," *International Journal of Computer Networks & Communications*, vol. 10, no. 1, pp. 13-33, 2018.

[11] W. Li, Z. M. Yang, Y. P. Chan, *et al.*, "A clustering algorithm oriented to intrusion detection," in *IEEE International Conference on Computational Science and Engineering*, pp. 862-865, 2017.

[12] I. V. M. D. Lima, J. A. Degaspari and J. B. M. Sobral, "Intrusion detection through artificial neural networks," in *Network Operations and Management Symposium*, pp. 867-870, 2008.

[13] 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.

[14] S. Rastegari, "Evolving statistical rulesets for network intrusion detection", *Applied Soft Computing*, vol. 33, pp. 348-359, Mar. 2015.

[15] R. Vijayanand, D. Devaraj, B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Computers & Security*, vol. 77, pp. 304-314, 2018.

[16] X. Zhao, C. Wang, "The improvements to snort intrusion detection system," *Journal of Xi'an Polytechnic University*, vol. 21, no. 6, pp. 859-863, Nov. 2007.

[17] X. Zhao, "Optimization of dynamic programming to the multimedia packets processing method for network intrusion detection system," *International Journal of Security and Its Applications*, vol. 9, no. 11, pp. 35-46, 2015.

[18] X. Zhao, "Research on a structure of the multimedia list oriented network intrusion detection system," *International Journal of Security and Its Applications*, vol. 10, no. 12, pp. 53-68, 2016.

[19] X. Zhao, "Dynamic self-adapting multimedia data processing method based on snort," *Computer System Application*, vol. 20, no. 4, pp. 211-213, 2011.

[20] X. Zhao, "The optimization research of the multimedia packets processing method in NIDS with 0/1 knapsack problem," *International Journal of Network Security*, vol. 17, no. 3, pp. 351-356, 2015.

[21] O. Marques, P. Baillargeon, "A multimedia traffic classification scheme for intrusion detection systems," in *International Conference on Information Technology and Applications*, pp. 496-501, 2005.

[22] S. Zander, G. ArmitagePractical, "Machine learning based multimedia traffic classification for distributed Qos management," *Local Computer Networks*, pp. 399-406, 2011.

[23] X. Y. Zhang, "Research of intrusion detection system dataset-KDD CUP99," *Computer Engineering and Design*, vol. 31, no. 22, pp. 4809-4812, Jan. 2010.

Biography

Xu Zhao is an associate professor in the School of Computer Science, Xi'an Polytechnic University, Shannxi, China. He received the M.S. degree from Xi?an Electronic Technology University, Xi?an City, Shannxi Province, China in 2007. He has developed several methods to deal with multimedia packets for network intrusion detection systems and is currently working on new optimization method with the help of artificial intelligence. He has some projects in research supported by provincial funds. His research interest is Network Security.

Guangqiu Huang is a professor and PhD supervisor at

Xi'an University Of Architecture And Technology. His major research interests include information security. He has completed 48 important scientific research projects.

Mr Reza Mousoli is a School Director of Stakeholder Engagement of School of Law, Criminal Justice and Computing of Canterbury Christ Church University of UK. His research Interests include Cyber Security, e-safety, Privacy and Confidentially.

A Novel Approach for Component based Application Logic Event Attack Modeling

Faisal Nabi[1], Jianming Yong[1], and Xiaohui Tao[2]
(Corresponding author: Faisal Nabi)
School of Management and Enterprise, University of Southern Queensland[1]
West St, Darling Heights QLD 4350, Australia
School of Sciences, University of Southern Queensland, Australia[2]
(Email: u1104061@umail.usq.edu.au)
(Received Aug. 4, 2019; Revised and Accepted Dec. 6, 2019; First Online Feb. 28, 2020)

Abstract

An Event that targets a particular system is required to identify through a novel approach of vulnerability modeling. Current research does not support Event Attack Modeling in component based application logic vulnerabilities. To find such vulnerabilities, it is important to identify the component that triggered the Event to exploit the system. This research proposes the Event Based Attack Modeling, especially in a scenario of component based software subversion logic attack category Business Application Logic. This will help to design and reuse of component from existing application's functional logic.

Keywords: Attack Modeling; CBS reuse; E-Commerce Application; Event Attack Method; Security Modeling

1 Introduction

Event based inter-component applications interact with each other through a passing message inter-communication mechanism [12]. This controlled by a distinct component that is called the event dispatcher, which performs its role as an intermediary between components where condition s are set for the system or application. In this process data communication is called an events that is generated from input communi-cation between components [11]. There are two more type of events, event parameters and event procedures that invoke the individual procedure called the event handlers. In an application, an event attack is occurred when any component of an application is mismatched with its design specification at integration stage. This may result of design fault, because of event-based interruption, which then can create a loophole to exploit the particular system, generated by an attack event during the inter-communication of event parameters [2].

The security vulnerability can arise in the environment that supports the event attack method. The source of the vulnerability can be based on object (component) that is able to generate the event send without any restriction and can be easily crafted into an event sequence for other objects (components) to circumvent the entire logic [9,27].

Event Interception is a phase of condition in which a victim object is identified and intercept the events destined to it. To be able to intercept the event sent to an object permits the attacker to breach the confidentiality of one direction of object (component) communication within the system [1,2,11,27]. In recent years there have been many application attacks based on logical flaws, such as logic flaw or design faults. There is a specific strategy that is required to deal with logical vulnerabilities, such logical attacks are classified as subversion attack. This attack is occurred because of logical flaw in design component based application and its interfaced based integration fault [4,7,25,27].

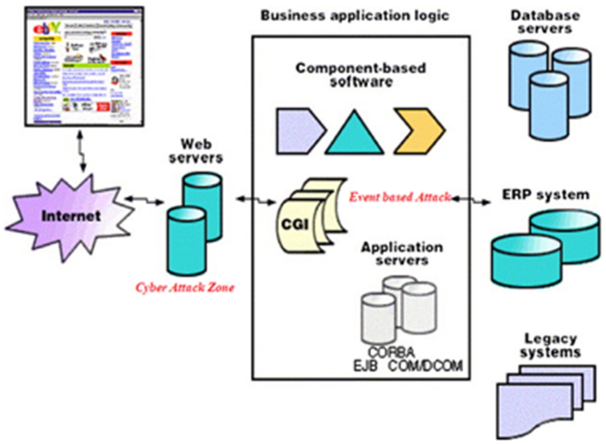


Figure 1: Component based application logic event attack scenario

Therefore, we classify this problem as an Event Attack View. In this case, specification refers to conventional attack, threat, vulnerability. This classifies the attack method, and attack model of identified vulnerability that is known as a subversion attack. In the field of cyber se-

curity Attack Event information is considered as at-tack related data that is derived from various sources. An at-tack event is defined as targeting assets by using attack method, which then exploits the functionality of application business process or circumvents the flow logic. It is very hard to detect the design flaw based vulnerabilities through traditional scanning tools; this is why such vulnerabilities never classified to deal with in terms of the application logic [5, 7].

In this research, we propose Event Based Attack Modeling for design flaw based vulnerability, called Subversion Attack (Component based application logic flaw) by using a Banking case study. The purpose of this re-search is to simplify the process of vulnerability modeling to understand the life cycle of vulnerability. This could help the developers while designing and reusing design specification of business components from existing application components and their underlining application logic. An Event Attack refers to a security problem that exploits the event based inter component communication model [5]. The definition of Event Attack: A malicious component that generates an event of circumvention in order to exploit the target's application logic or functionality. This intercepts communication by forcing the targeted component to send back an inappropriate call or calling away from application functional logic [5].

2 Problem Statement

The focus of this research is to analyze the Event at-tack model and the Subversion attack that falls in the category of business logic vulnerability. Specially considering the security breach scenario real life case study related to Barclay bank, as well as the re-usability design description of component.

The research question, how can Event Attack Modeling simplify the application logic vulnerability, subversion attack? This question is answered by the example of real time case study research method, using Event At-tack Modeling technique.

This real-life case study is a good example of a design flaw in application logic due to the reuse of a component caused component subversion. In this example, the developer reused the same component that was already incorporated in the registration functionality elsewhere within the application, violating the assumptions of the component developer. This mistake lead to the introduction of an application-level flaw that allowed an attacker to access another client's bank accounts. The approach taken to be analyzed, this problem is one that the Event At-tack Modeling Technique will be able to helpful to detect design flaws and/or fault free component-based application logic in the middle tier of the n-tier architecture as depicted in Figure 1.

2.1 Research Philosophy

The research philosophy is taken as applied science that is basically an application of existing scientific knowledge to practical applications such as technology, concerning the theory of Event of inter component-communication model. It uses theory, knowledge, method and technique for a particular state of the art [28]. This discussion about Component-based State of the Art in relation to the philosophy of its application & design pattern. The research philosophy also defines and investigates about state of the art technology in Event interaction between the component software de-signs, which is adopted from an applied science philosophy to formulate a solution for business logic vulnerability. In this process, it is very important to understand that design question in the light of research philosophy, can help to conduct the research in the field of Attack Modeling & Security domain by ensuring that research-er's work is going in a right direction and their work is rigorous and insightful.

2.2 Research Gap

In the light of current research and recently studied literature review, [6, 14, 18, 21] and [17] in the domain of cyber and network vulnerability modeling. The research Gap clearly finds an interest to improve the business logic security, specially "Design Flaw" in a service oriented e-commerce applications, that is composed with integrated components. The research gap identified the significance of application logic vulnerability class and category "Subversion attack" cause of Design Flaw, because automated vulnerability analysis and detection tools cannot detect it. This is reason why such vulnerabilities are always oversighted by the application developers. The developers are always keen to reuse existing component core logic from current business logic of the system. This may often cause of mistake while integrating component code solution and designing new functionality.

2.3 Research Design and Method

This research is based on exploratory method where no scientific foundation is available for supporting techniques. The current research and literature review highlights the gap between the current approach and previously designed models or frameworks for logical vulnerabilities. Therefore, we have proposed (Event Attack Modeling) such a technique that could deal with application level logic vulnerabilities. This would help to detect early design faults at the time of integration of components and design fault free new applications. The re-search design also follow previous modeling techniques to justify the newly proposed technique. This simplifies the problem detection process and method.

2.4 Current Approaches in Attack Modeling

There have been several techniques used for vulnerability modeling. These techniques are Attack Graph [26], Attack-Vector [22], Attack-Surface [22], Diamond model [13], OWASP's threat model [13] and Kill Chain [15]. Each technique has its own properties and speciality to identify and model the attack process path way through out the system and network. For example, Attack Graph technique is used for network related vulnerability and system exploitation modeling based on scenario of security issues. Through this technique one can identify the process and pathway of security breach cause within the network as shown in Figure 2.

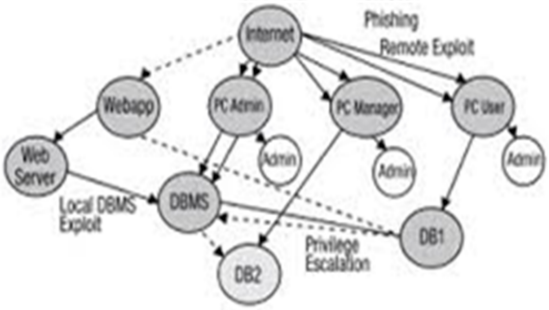


Figure 2: Attack graph with attack path against system

3 Studying Case Profile & Event Attack Modeling

This real life case is a good example of a design flaw in application logic due to the reuse of a component caused component subversion. In this example, the developer reused the same component that was already incorporated in the registration functionality elsewhere within the application, violating the assumptions of the component developer. This mistake leads to the introduction of an application-level flaw that allows an attacker to access an other client's bank accounts (component code Figure 3).

```
class CCustomer
{
 String firstName;
 String lastName;
 CDoB dob;
 CAddress homeAddress;
 long custNumber;
... }
```

Figure 3: *C* customer component code

3.1 Component Application Logic Design Fault

The registration functionality incorporated with the *CCustomer* component that consist of "(use case logic + *Process* and *Entity* Type Logic)" within the application, including core functionally. This process allows the user to authenticate and grant access to the application components such as "My Account component", "View Balance component", "Funds transfers component", "Select Bank Account component, Debit Credit component and other information component. After having authenticated user itself to the application through the registration process, the same Object instantiate and saves in the session key information related to the identity. The components of application within functionally referenced information related to the $^{\star}CCustomer(Component)^{\star}$ object in order to carry out its actions because the $^{\star}CCustomer(Component)^{\star}$ object is candidate component (*Process* and *Entity* Type logic) within the majority of application — for example, account details shown on the main page of the user was generated based on the customer unique number that contained within this component. In the way composition or reuse of the component, code was already used within the application. It clearly shows that the developer assumption leads to a flaw in the reuse of application logic design. This caused the birth of a vulnerability to subversion attack on application business logic. It was a serious mistake and subtle to detect and exploit.

3.1.1 Class of Vulnerability

The "*Subversion Attack*" characterization of vulnerability flaw falls under the application logic, and attack method is to exploit the workflow of business logic, this process subvert business process. At implementation level it is classified as design logic flaw, which then finally characterized as "Subversion of logic" attack.

Subversion of logic. Class: Programme logic flaw;

Server application: (Target agent);

Attack method: (Exploit the work flow);

Subvert application logic: (Attack cause);

Implementation level: (Application design logic flaw classification);

Vulnerability: Subversion of logic.

Therefore, we modeled the Event oriented subversion life cycle that displays the logic diversion of business logic in a small chain of inter-component based communication application model, caused by CBS Flaw.

The above mentioned Figure 4 displays an event attack model scenario, class is subversion attack that falls under business application logic vulnerability, based on component based software that may be flawed in CBS. This fault may have effects on service calls and flow of the function that depends on event based call to other

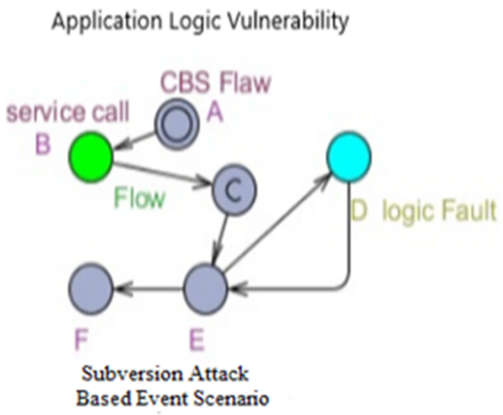


Figure 4: Subversion attack event scenario

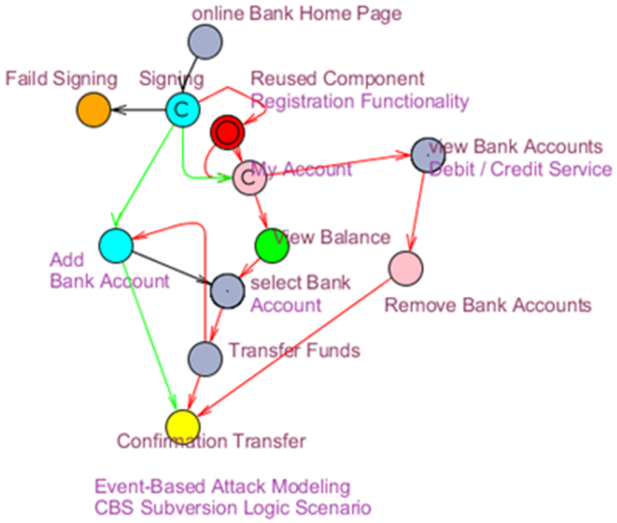


Figure 5: Event attack subversion logic scenario

objects within the system. As it is shown in the above Figure 4, C is condition that must correspond to component D before processing to normal application logic flow to proceed the E. Therefore, D component is a logic fault that does not let the service flow according to normal flow of CBS call service,this is reason why such faults cannot be detected by automated code & system vulnerability scanning tools,and such faults or flaws fall under the classification of logical vulnerability.

3.2 Case Scenario Based Experimental Study

We have further investigated the scenario of this attack keeping in view the above mentioned example related to a security breach of Bank case study. This is caused by a logical design flaw within the system while reusing component from existing application logic. This is called "Subvert Event based Attack" on the banking application. The developers always oversight such attacks on the application's business logic, even though it is a serious vulnerability. It is hard to detect through code scanning and automated detection tools. Therefore, such a technique is required that could simplify the projection of this vulnerability, through the approach of Event based Attack modeling. The proposed technique seems to be a new and effective technique for early detection of such attack at design level of application.

The above-mentioned Figure 5 displays the complete life cycle of the Event Attack Model. In the model C indicates to a condition, If **sign-in**, Pass log in to **My Account** Condition to allow access into the system, **Else** Failed **sign in**. This is the general case of scenario system logic for sign in. However, the major mistake is done by the application developer of the banking system reused same component that was already incorporated in the registration functionality elsewhere within the application. This mistake causing subversion of logic and by pass the condition that is set on **My Account (component)** this violated the assumptions of the component developer and caused the system under attack. This attack also subvert the other components of the application service flow as shown in Figure 5. Any intrusion detection tool cannot detect this sort of attack known as a class of application logic subversion attack. Therefore security scanning automated software, fail to discovery and un-automate this class of vulnerability. The reused component in the application is spotted in **Red Color**, which reflects the service flow diversion and allow an Event to trigger a logical attack by passing session and controls security mechanism of an application related to other service components as displayed in Figure 5. Therefore, above model cycle of an attack is modeled through a Event Attack Modeling technique in scenario of Component-based Software subversion logic Fault.

3.3 Theoretical Analysis of Proposed Approach

In the light of cyber attack theory a successful attack relies on information to be processed by attacker, in case of when an attack is underway and it is measured by modifying as a result related to attack. Therefore, in-formation is a most important element of any cyber at-tack theory [25].

As, it is confirmed that in the theory of cyber attack, first attack is defined and then attacker knowledge related to information parameters and configuration parameters are derived in order to mitigate the system from potential damage [10].

Therefore we formalized the theory of cyber attack into proposed approach event attack modeling.In this pro-cess, first identified the attacker and then measured the attack information parameters, through that an event is occurred as a fault logic, service component triggered to flow diversion and allow an Event trigger by passing session and controls security mechanism. This is demonstrated through scenario based event attack modeling Figure 5 that helped to diagnose the vulnerability life-cycle. This

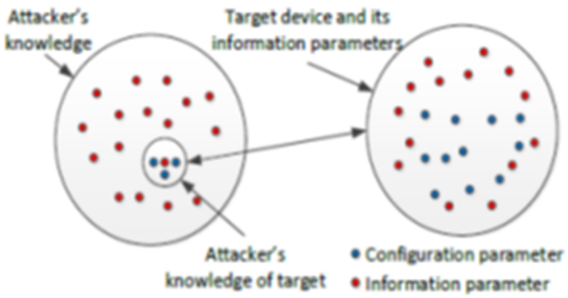


Figure 6: Cyber attack theory model

gives the knowledge related to information attack parameters, and component configuration parameters that decides the attack vector related to vulnerability of application logic class (subversion logic attack).

Therefore, it is concluded that above mentioned technique is very useful for attack modeling in the light of cyber attack theory.

3.4 Systematical Comparison of the Proposed Scheme

The current approaches of attack modeling are based on attack graph and vector modeling techniques [22, 26], these techniques models focus on the network or system vulnerability based modeling that deals with the different attacks targeting the network [10], but the lack of software application scenario based modeling.In this, scenario an approach is immanent for application based vulnerability modeling technique. Therfore, the proposed scheme is presented, event based attack modeling that targets the service component triggered to flow diversion of application logic in component-based system. The proposed scheme is comparably sounder as compare to any other modeling technique for software based application and its core logic flow.

3.5 Discussion

We have seen that the proposed technique is very helpful in detecting the event that triggered the subversion attack within the application and its component at the integration level, which clearly depicts the vulnerability and its effects on other components of the application and underling business logic. We also have evaluated the other techniques such as Attack Graph and Attack Vector. The Attack Graph is use to identify the vulnerability in the networks and system, and Attack Vector can provide the path way projection through hacker exploitation attempt which targets the network servers by payload or malicious input. It is also modeled through Attack Vector Modeling technique. It has been noticed that none of these techniques meet the requirement of logical attack modeling and simulation [18].

Where as proposed technique is useful to model the case scenario of banking application through Event Based Attack modeling. That is spotted in red color the component with fault service flow, calling C condition **My Account** component within the application that cause exploitation.

4 Related Work

There are numbers of approaches target the security in event based inter-component applications [3, 19, 23, 24]. For example, Simeon *et al.* [27] took into account the security vulnerabilities in event-based applications and systems, explained the conditions that can be made of them,in result of inter-communication fault. In simple term, current security solutions more rely on encryption, static code analysis, and runtime ACL techniques. Whereas, on the other hand, there have been many techniques adopted to attack modeling such as the Diamond Model [13], Attack Tree [20], Attack Vector [22], Attack Surface [16], Kill Chain [15] and Attack Graph [26]. However, all of these techniques fail to ad-dress the logical vulnerabilities detection or modeling framework, because these techniques are network vulnerability modeling and address the network security issues related to the system. Therefore, such a technique needs to introduce that can deal with missing gap between application and system level vulnerability modeling. This will fill the research gap related to logical vulnerabilities in application logic (Component-based Software) [8].

5 Conclusions

Attack modeling is a most useful technique in ana-lysing the attacks and early mitigation of the problem. This is why many techniques are introduced to deal with the attack modeling in the system network domain. The logical vulnerabilities are flaw in design or fault in logic. It is hard to detect and modeled. Therefore such a technique is required that could deal with the logical flaw based vulnerability. In this paper, we have introduced a novel approach of modeling called "Event Attack Modeling" that used Uppaal Tool to model the vulnerability and its attack flow through attack-triggered component within the application in real time scenario. This will help the developers design their application free from logical flaws and design faults, while reusing design specification of component from existing application.

References

[1] A. A. Al-khatib, W. A. Hammood, "Mobile malware and defending systems: Comparison study," *International Journal of Electronics and Information Engineering*, vol. 6, no. 2, pp. 116–123, 2017.

[2] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen, J. Disso, "Cyber-attack modeling analysis techniques: An overview," *The 4th International*

Conference on Future Internet of Things and Cloud Workshops, 2016. DOI: 10.1109/W-FiCloud.2016.29.

[3] L. Aniello, R. Baldoni, C. Ciccotelli, G. A. D. Luna, F. Frontali, and L. Querzoni, "The overlay scan attack: Inferring topologies of distributed pub/sub systems through broker saturation," in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems (DEBS'14)*, pp. 107—117, 2014.

[4] A. Anurag, "Network neutrality: Developing business model and evidence based net neutrality regulation," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 1–9, 2015.

[5] Bank of England, "An introduction to cyber threat modelling", Industry report, Bank of England Publication, 2016. (`https://www.cyentia.com/library-item/an-introduction-to-cyber-threat-modelling/`)

[6] M. Bentounsi, S. Benbernou, M. J. Atallah, "Security-aware business process as a service by hiding provenance," *Computer Standards & Interfaces*, vol. 44, pp. 220–233, 2016.

[7] BSIMM, "Attack models with bsimm frameworks," 2016. (`https://www.bsimm.com/framework/intelligence/attack-models/`)

[8] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, pp. 80, 2011.

[9] S. Islam, H. Ali, A. Habib, N. Nobi, M. Alam, and D. Hossain, "Threat minimization by design and deployment of secured networking model," *International Journal of Electronics and Information Engineering*, vol. 8, no. 2, pp. 135–144, 2018.

[10] S. Jajodia and S. Noel, *Advanced Cyber Attack Modeling, Analysis, and Visualization*, George Mason University, Mar. 2010. (`https://csis.gmu.edu/noel/pubs/2009_AFRL.pdf`)

[11] N. J. Kim, M. S. Gong, G. S. Lee, "An attack-target-method schema for cyber attack event database," *IEEE International Conference on Electronic Information and Communication Technology (ICEICT'16)*, 2016. DOI: 10.1109/ICEICT.2016.7879705.

[12] Y. K. Lee, D. Nam, N. Medvidovic, *Identifying Inter-Component Communication Vulnerabilities in Event-based Systems*, Technical Report: USC-CSSE-17-801, 2016.

[13] X. Lin, P. Zavarsky, R. Ruhl, and D. Lindskog, "Threat modeling for CSRF attacks," in *IEEE 16th International Conference on Computational Science and Engineering*, vol. 3, pp. 486-–491, 2009.

[14] A. K. Luhach, S. K. Dwivedi, C. K. Jha, "Designing and implementing the logical security framework for e-commerce based on service oriented architecture," *International Journal on Soft Computing (IJSC'14)*, vol. 5, no. 2, 2014.

[15] P. K. Manadhata, J. M. Wing, "An attack surface metric," *IEEE Transactions on Software Engineering*, vol. 37, no. 3, pp. 371-–386, 2011.

[16] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl, "Dark clouds on the horizon: Using cloud storage as attack vector and online slack space," in *USENIX Security Symposium*, pp. 65-–76, 2011.

[17] F. Nabi, "Designing a framework method for secure business application logic integrity in e-commerce systems," *International Journal of Network Security*, vol. 12, no. 1, pp. 29-–41, Jan. 2011

[18] F. Nabi and M. M. Nabi, "A process of security assurance properties unification for application logic," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 40–48, 2017.

[19] F. Petroni, L. Querzoni, R. Beraldi, and M. Paolucci, "Exploiting user feedback for online filtering in event-based systems," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC'16)*, pp. 2021-–2026, 2016.

[20] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proceedings of the 1998 Workshop on New Security Paradigms*, pp. 71-–79, 1998.

[21] L. Seinturier, P. Merle, R. Rouvoy, D. Romero, V. Schiavoni, and J. B. Stefani, "A componentbased middleware platform for reconfigurable service-oriented architectures," *Software Practice and Experience*, vol. 42, no. 5, pp. 559—583, 2017.

[22] B. Schneier, "Attack trees," *Dr. Dobb's Journal*, vol. 24, no. 12, pp. 21—29, 1999.

[23] B. Shand, P. Pietzuch, I. Papagiannis, K. Moody, M. Migliavacca, D. Eyers, and J. Bacon, "Security policy and information sharing in distributed event-based systems," *Reasoning in Event-Based Distributed Systems*, pp. 151—172, 2011.

[24] M. Srivatsa, L. Liu, and A. Iyengar, "Event-guard: A system architecture for securing publish-subscribe networks," *ACM Transactions on Computer Systems (TOCS'11)*, vol. 29, no. 4, pp. 10:1—10:40, Dec. 2011.

[25] A. Tayal, N. Mishra and S. Sharma, "Active monitoring & postmortem forensic analysis of network threats: A survey," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 49–59, 2017.

[26] United States. Joint Chiefs of Staff, *Joint Tactics, Techniques, and Procedures for Joint Intelligence Preparation of the Battlespace*, 2000. (`http://purl.access.gpo.gov/GPO/LPS49610`)

[27] S. (simos) Xenitellis, "Security vulnerabilities in event driven systems," in *Proceedings, Security in the Information Society: Visions and Perspectives*, pp. 147–160, 2001.

[28] A. Yaghmaie, "How to characterise pure and applied science," *International Studies in the Philoso-Phy of Science*, vol. 31, no. 2, pp. 133–149, 2017.

Biography

Faisal Nabi is a PhD researcher at University of Southern Queensland.He has also received Honorary PhD in Computer Science from Brock University St. Catharines, Ontario, Canada. Faisal's research interests are e-commerce security and software security.

Jianming Yong is Professor of school of information systems.He has received his PhD from SwinburneUT.He is also member of IEEE professional.His research areas are Cloud Computing, Big Data Security and Privacy, Data Integration, Workflow systems, Information system security, Network management, Web service for SMEs, Digital Identity Management.

Xiaohui Tao is Associate Professor in School of Sciences, University of Southern Queensland, Australia. His research interests include Natural Language Processing, Text Mining, Knowledge Engineering, and Health Informatics. During his research career, Tao has gained a wealth of knowledge and experience in dealing with massive datasets and delivering solution to complex research problems, and made many contributions to Ontology Learning, Web Intelligence, Data Mining, and Information Retrieval. His research results have been published in 90+ refereed papers, many of them are on highly ranked journals such as IEEE TKDE, KBS, PRL and conferences such as ICDE, PAKDD and CIKM. He has been a Program Chair of many International Conferences and Workshops.

Anonymous Transaction of Digital Currency Based on Blockchain

Yang Liu, Mingxing He, and Fangyuan Pu
(Corresponding author: Mingxing He)

School of Computer and Software Engineering, Xihua University
999 Jin Zhou Road, Jin Niu District, Chengdu, 610039, China
(Email: he_mingxing64@aliyun.com)

(Received Aug. 3, 2018; Revised and Accepted Jan. 24, 2019; First Online Oct. 14, 2019)

Abstract

Blockchain can be seen as a shared database, and keep all data public and traceable. Everyone is accessible to the data recorded on the blockchain, which brings the risk of privacy leakage. When digital currency transactions are performed on the blockchain, users may not want to reveal their real identities. Therefore, it is particularly important to preserve the identity privacy of users. To solve the problem, we present an anonymous transaction scheme of digital currency to ensure the anonymity of the sender and receiver. we design a linkable ring signature algorithm based on elliptic curve cryptography (ECC) to conceal the real identity of sender and check double-spending. It is intermediate address that is used for concealing the real identity of receiver. Furthermore, we utilize a agency to reduce computational burden for receiver. Throughout the transaction process, the real identities of two sides are not disclosed to others, ensuring anonymous transaction.

Keywords: Anonymous Transaction; Blockchain; Digital Currency; Linkable Ring Signature

1 Introduction

The essence of blockchain is a huge distributed database without unified manager. The data is stored in blocks, and all blocks are linked in the form of chain structure. All records on the blockchain are public and traceable. Users can believe in records on the blockchain, and not have to trust the third parties such as banks or governments. Blockchain is mostly used for storing transactions information to keep traceable and avoid central domination. It has wide application prospect in various fields, especially digital currency.

Bitcoin [16] is regarded as the first digital currency based on blockchain, and also the most typical and successful application of blockchain technology. Bitcoin transactions are verified by all nodes on the blockchain and can never be falsified. All digital currencies based on blockchain that appear after Bitcoin are derived from Bitcoin. Therefore, our scheme can also be regarded as being based on Bitcoin.

To verify transactions without relying on the third parties, blockchain must build consensus among distributed nodes [8]. Therefore, all records on the blockchain must be public and can be viewed by any node, which leads to plenty of private information being exposed [10].

In response to the demand for privacy-preserving, there are some schemes have been proposed. Most schemes utilize mixing coins, being divided into centralized and decentralized. The idea of mixing coins originates from the paper published by Chaum [3]. It is used to achieve anonymous communication between the two sides through the intermediary transferring information, so that the attacker is unable to accurately determine whether the two sides communicate. The mixing coins in the transaction of digital currency draws on this idea, and confuses the transaction contents without changing the transaction results, hiding the relationship between input and output.

Some centralized mixing coins are operated by the third parties. Many companies offer mixing coins service to make money, such as Bitcoin Fog and Bitlaunder. Users can enjoy the mixing coins service after paying the service fee, but this approach carries the risk of funds being stolen by the third parties. There are some centralized mixing cions algorithms by utilizing a central node to execute. Mixcoin [2] adds an accountability mechanism to expose theft. Blindcoin [24] is optimizing of Mixcoin, and use blind signature to hide the relationship between input and output. Then, ShenTu *et al.* [22] propose a mixing coins scheme based on blind signature and it increases computational efficiency on the basis of ensuring anonymity. Recently, Liu *et al.* [13] propose a mixing coins scheme based on ring signature with centralized mixing server. However, centralized mixing coins has a distinct shortcoming and the central node may leak the information about mixing coins.

The decentralized mixing coins has been first proposed by Gregory in CoinJoin [14]. It combines multiple trans-

actions into one transaction to provide anonymity for users, which requires users to execute mixing coins autonomously. CoinShuffle [18] designs a shuffle protocol to improve CoinJoin, and requires participants to be online at the same time. Thus, it has low efficiency and is vulnerable to Dos attack. Subsequently, Xim [1] utilizes announcements on the blockchain for aggregating users who want to take part in mixing coins, and is able to resist Dos attack, but it only supports two-party mixing coins and has low efficiency. SecureCoin [7] improves security and efficiency over the CoinShuffle. Coinparty [27] makes use of secure multi-party computation to ensure the availability of mixing coins when there are malicious processing. In short, mixing coins need numerous users to participate and cooperate with each other, so there is still a risk of information leakage.

Many anonymous digital currencies also provide a new way for privacy-preserving. We take Zerocash and Monero as example in the following. Zerocash [20] inherits the thought of Zerocoin [15] scheme, forming the best anonymous digital currency. It converts the user's coins into equivalent commitment. When users want to spend the funds, they utilize zero-knowledge proofs to prove that the funds belong to themselves and have not been spent. It ensures unlinkability of transactions, but has a bottleneck in efficiency. The core of Monero is CryptoNote [19] protocol. It ensures anonymity of transaction by ring signature based on non-interactive zero-knowledge proofs, so it is quite complicated in calculation.

Recently, some new privacy-preserving schemes are proposed about blockchain. Heilman *et al.* [5] present the micropayment channel networks and combine blind signature with smart contract, to achieve the anonymity for Bitcoin transactions. Kosba *et al.* [9] present Hawk scheme, and it combines zero-knowledge proofs with secure multiparty computation to achieve privacy-preserving about smart contract on the blockchain. Yuan *et al.* [26] propose a new ring signature scheme for the transactions on blockchain based on aggregate signature and ECC. When the transaction contains multiple inputs and outputs, it can achieve both hiding the amount of the transactions and constant-size signature, but it is only aimed at privacy-preserving of the transactions, without regarding to double-spending. There are also encryption schemes about identity, such as Liu *et al.* [12] propose an anonymous identity-based encryption scheme, and it improves the SKOS scheme [21] and proves its security under l-computational Diffie-Hellman assumption.

Based on blockchain, we propose a anonymous transaction scheme about digital currency without the third parties. There are two crucial technologies, that is, linkable ring signature and intermediate address. We design a new linkable ring signature algorithm based on ECC and make use of the advantages of ECC, that is, high security and fast processing. Linkable ring signature is used for concealing the real identity of sender to ensure anonymous payment, and also used for checking double-spending. Namely, double-spending means that someone spends the same money twice, and it especially occurs in the transactions of digital currency. Intermediate address can be regarded as a virtual address, which is generated by sender. Intermediate address is used for concealing the real identity of receiver to ensure anonymous receiving. Nobody has the ability to judge who the intermediate address belongs to, except the sender and receiver. Our scheme is able to preserve simultaneously the identity privacy of two sides and achieve anonymous transaction.

The content of this paper is organized as follows: Section 1 introduces briefly some background knowledge and major components about this paper. Section 2 introduces the preliminaries. Section 3 proposes a linkable ring signature algorithm based on ECC. Section 4 introduces details of our anonymous transaction scheme. we present analysis about our proposed scheme in Section 5. The last section concludes this paper.

2 Preliminaries

2.1 Ring Signature

In 2001, ring signature was first proposed by Rivest, Shamir and Tauman [17]. The signer aggregates an arbitrary set of users (their public keys) with his own private key to form a ring structure in a certain rule. Ring signature provides an anonymous way to sign message without revealing any identity information. The verifiers can be convinced that the signature comes from this group. However, nobody has the ability to identify who is the real signer unless the real signer exposes himself.

In some cases, While ensuring anonymity, we also need to know whether two signatures are signed by the same signer. To solve this problem, Liu *et al.* [11] first propose the concept of linkable ring signature. It has the characteristic of linkability compared to ordinary ring signatures. It can prove whether two signatures are signed by the same signer by means of adding an linkable tag to the signature. If two signatures have the same linkable tag, it means that they are signed by a signer for the same message, so they are linked. Base on this notion, Gu *et al.* [4] present a fully traceable certificateless ring signature scheme. However, it is not enough efficient in judging linkability of signatures.

In this paper, we design a new linkable ring signature algorithm, and utilize the linkability to check double-spending in the transactions of digital currency. To ensure the uniqueness of every linkable tag, we add the private key of signer and the signature of previous transaction to generate the linkable tag.

2.2 Elliptic Curve Cryptography

The elliptic curve cryptography (ECC) is an important branch of the public key cryptosystem (PKC), and its security is based on the difficulty of elliptic curve discrete

logarithm problem [23,25]. Compared with the RSA public key cryptosystem, ECC has less computation, faster processing, and less storage space and transmission bandwidth [6]. The Bitcoin also selects ECC as the encryption algorithm. Our scheme is based on ECC and completely compatible with Bitcoin.

We briefly introduce the principle of ECC as following. Consider $A = aP$, where A, P are the points on the elliptic curve E, q is the order of P, and a is an integer less than q. According to the addition rule on the elliptic curve, given a and P, it is easy to compute A, but conversely, given A and P, it is very difficult to find a. Therefore, we usually take a as private key, A as public key.

3 Linkable Ring Signature Based on ECC

Let E represent an elliptic curve defined on a finite field $GF(p)$. Let G be a group with generator P on elliptic curve E. Let q represent the order of P, where q is a large prime number. $L = \{K_1,K_2,\cdots,K_n\}$ represents the list of n public keys. For $i = 1,2,\cdots,n$, each user i has a distinct public key K_i and private key k_i such that $K_i = k_iG$, where $k_i \in [1, q-1]$. It is worth noting that μ represents the signature of previous transaction in our anonymous transaction scheme, that is to say, it stands for the source of the funds in the current transaction, so it is public and unique on the blockchain. We define two cryptographic hash functions :

$$H_1 : \mathbb{Z}_q \to G \text{ and } H_2 : \{0,1\}^* \to \{0,1\}^d.$$

3.1 Signature Generation

Given message $m \in \{0,1\}^*$, list of public keys $L = \{K_1,K_2,\cdots,K_n\}$. Let j stand for the real signer, and the public key K_j corresponding to private key k_j, where $1 \le j \le n$. User j generates a linkable ring signature $\sigma(m)$ as following steps:

1) Compute $h = H_1(\mu)$, $W = \mu h$, and $U = \mu G = (x, y)$.
2) Compute $\tilde{h} = xk_jh$, and $(\tilde{h}, U)$ is regarded as linkable tag.
3) Pick $t \in_R [1, q-1]$, and compute $T = tG$, $T' = th$, and $c_{j+1} = H_2(m, T, T')$.
4) For $i = j+1, \cdots, n, 1, \cdots, j\text{-}1$, pick $s_i \in_R [1, q-1]$, and compute $c_{i+1} = H_2(m, s_iG + xc_iK_i + c_iU, s_ih + c_i\tilde{h} + c_iW)$, and take $c_1 = c_{n+1}$.
5) Compute $s_j = t - xc_jk_j - c_j \ \mu \pmod q$.
6) Finally, construct the signature

$$\sigma(m) = \{L, c_1, s_1, \cdots, s_n, h, \tilde{h}, U\}.$$

3.2 Signature Verification

The verifier verifies the validity of the signature $\sigma(m)$ as follows:

1) Extract μ from blockchain, and compute $W = \mu h$.
2) Extract x from U, for $i = 1,2,\cdots,n$, compute $T_i = s_iG + xc_iK_i + c_iU$, $T_i' = s_ih + c_i\tilde{h} + c_iW$, and then $c_{i+1} = H_2(m, T_i, T_i')$ if $i \neq n$.
3) Check whether $c_1 \stackrel{?}{=} H_2(m, T_n, T_n')$. If yes, accept. Otherwise, reject.

3.3 Linkability

Given two signatures,

$$\sigma'(m') = \{L',c_1',s_1',\cdots,s_n',h',\tilde{h}',U'\},$$
$$\sigma''(m'') = \{L'',c_1'',s_1'',\cdots,s_n'',h'',\tilde{h}'',U''\},$$

the verifier checks if $\tilde{h}' \stackrel{?}{=} \tilde{h}''$ and $U' \stackrel{?}{=} U''$. If two equations both hold, it means that the linkable tag of two signatures are the same. The verifier can conclude that $\sigma'(m')$ and $\sigma''(m'')$ are linked, namely, they are generated when a user signs two transactions m' and m'' that include the same money. Otherwise, the verifier concludes that two signatures are not linked. Therefore, two signatures are linked if and only if their linkable tags are the same.

In our anonymous transaction scheme, the linkable tag is generated by utilizing the private key k_j of signer and the signature μ of previous transaction. Because k_j is unique for the signer, and μ is also unique for the funds in the current transaction, the linkable tag is certainly unique. Therefore, two linkable tags must be the same when a user signs two transactions that include the same money, and it indicates that the user spends the same money twice, that is to say, double-spending. Consequently, we check double-spending in our scheme by utilizing the linkability of linkable ring signature.

3.4 Properties of the Signature

This linkable ring signature algorithm based on ECC inherits the characteristics of both ECC and linkable ring signature, like high security, anonymity, linkability, and unforgeability.

1) anonymity. User signs the transaction by utilizing the linkable ring signature. The verifier can be convinced that the signature comes from the group. Since every member of the group has equal position, anyone of the group may generate the signature $\sigma(m)$. Therefore, nobody has the ability to identify who is the real signer unless the real signer exposes himself.
2) linkability. Both the private key k_j of signer and the signature μ of previous transaction are unique, so the linkable tag $(\tilde{h}, U)$ is certainly unique. If two

signatures have the same linkable tags, and it means that two signatures are linked. If two signatures are linked, they have the same linkable tags. Therefore, two signatures are linked if and only if their linkable tags are the same. When a user signs two transactions that include the same money, two signatures are linked and have the same linkable tags. It indicate that the user spends the same money twice, that is to say, double-spending.

3) unforgeability. Firstly, this linkable ring signature algorithm is based on ECC, and its security is based on the difficulty of elliptic curve discrete logarithm problem. Secondly, the signer generates a valid linkable ring signature $\sigma(m)$ and must use his own private key. If the attacker wants to forge the signature $\sigma(m)$, he must solve the elliptic curve discrete logarithm problem and also hold the real signer's private key. It is difficult to solve the elliptic curve discrete logarithm problem, and he is also unable to obtain the real signer's private key. Therefore, nobody can forge the signature $\sigma(m)$.

4 Transaction Scheme

4.1 Scheme Overview

We present the details of our anonymous transaction scheme in this section. This scheme has a virtual intermediate address and three participants, that is, sender Alice, receiver Bob and agency Carlo. The overview of our anonymous transaction scheme is shown in Figure 1. The symbols used in our scheme are listed in Table 1.

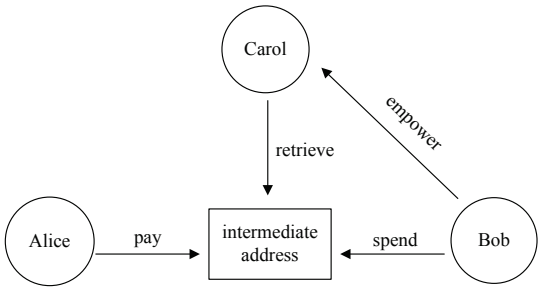


Figure 1: Overview of the scheme

In order to ensure anonymous simultaneously of sender Alice and receiver Bob, they have three public and private key pairs in our scheme as shown in Table 2, that is, a main public and private key pair and two standby public and private key pairs. The main public and private key pair is used for generating linkable ring signature $\sigma(m)$. Two standby public and private key pairs are used for generating the intermediate address and empowering the agency Carlo to retrieve transactions. The intermediate address of our scheme is slightly similar to the stealth address of CryptoNote [19]. However, the difference is that our scheme uses three public and private key pairs and adds an agency Carlo to help receiver Bob retrieve transactions.

It is worth noting that agency Carlo can only be regarded as an assistant to receiver Bob for retrieving transactions. Since it has heavy computational burden to retrieve transactions on the blockchain, our scheme adds an agency Carlo to help receiver Bob retrieve transactions. When Bob empowers Carlo to retrieve transactions on the blockchain, Bob sends anonymously a part of keys to Carlo with a sum of agency fee. Carlo retrieves transactions belonging to Bob on the blockchain. Certainly, Bob can also retrieve it by himself if he has plenty of computing resources. Emphatically, we add the agency in our scheme only when Bob has limited computing resources. The agency is based on reputation for the purpose of making money in our scheme. If the agency has fraudulent behavior, his reputation will be damaged leading to poor business.

Table 1: Symbol and description

Symbol	Description
p	Order of the generator P
H	Hash function
R	Payment evidence
L	The list of public keys
Y	Intermediate address
Y^*	The address computed by agency
K^*	Sum of the Bob's standby public keys
$\sigma(m)$	Linkable ring signature for message m
$(\tilde{h}, U)$	Linkable tag

Table 2: Key and description

Key	Description
$K_{a1} = k_{a1}G$	K_{a1} is the main public key of Alice. k_{a1} is the main private key of Alice.
$K_{a2} = k_{a2}G$ $K_{a3} = k_{a3}G$	K_{a2} and K_{a3} are the standby public keys of Alice. k_{a2} and k_{a3} are the standby private keys of Alice.
$K_{b1} = k_{b1}G$	K_{b1} is the main public key of Bob. k_{b1} is the main private key of Bob.
$K_{b2} = k_{b2}G$ $K_{b3} = k_{b3}G$	K_{b2} and K_{b3} are the standby public keys of Bob. k_{b2} and k_{b3} are the standby private keys of Bob.

4.2 Payment Protocol

Alice intends to initiate a payment of digital currency to Bob, and generates a transaction including receiver's address, payment amount, payment evidence, time-stamp, sender's signature. Let m represent the transaction information. In fact, Alice pays a sum of money to intermediate address instead of the real address of Bob.

Alice generates a transaction as following steps:

1) Obtain the main public key K_{b1} and two standby public keys K_{b2} and K_{b3} of Bob from blockchain.

2) Pick $r \in_R [1, q-1]$, and compute $R = rG$, and compute intermediate address $Y = H(rK_{b1})G + K_{b2} + K_{b3}$. Y is specified as receiver's address. R is specified as payment evidence, which is used for resisting to denial of receiver.

3) Send R to Bob.

4) Use the main public key K_{a1} and main private key k_{a1} to construct the linkable ring signature $\sigma(m)$ by the signature generation algorithm of Section 3.1. $\sigma(m)$ is specified as sender's signature, and it includes linkable tag $(\tilde{h}, U)$.

5) The time-stamp is generated automatically by blockchain to record current time.

6) Finally, broadcast the transaction anonymously to blockchain.

4.3 Verification Protocol

Our transaction scheme is based on blockchain, and it is based on Bitcoin for the process of verifying transaction. Therefore, we omit the details about verifying Bitcoin transaction, and only describe the part of our design.

When the verifier receives the transaction, he firstly checks double-spending then verifies signature $\sigma(m)$. Our scheme requires that all nodes on the blockchain store a spent-list including the linkable tag of every transaction to check double-spending. Every node verifies the transaction as following steps:

1) Extract linkable tag $(\tilde{h}, U)$ from signature $\sigma(m)$.

2) Check $(\tilde{h}, U)$ whether exist in the spent-list. If yes, indicate double-spending and reject this transaction. Otherwise, go on verifying $\sigma(m)$ according to the signature verification algorithm of Section 3.2.

3) If $\sigma(m)$ is verified being valid, this transaction is also valid and can be recorded in blocks, then is added to the blockchain. Otherwise, reject this transaction.

4.4 Retrieval Protocol

Bob empowers Carlo to retrieve transactions on the blockchain When his computing resources are limited. Bob sends anonymously a part of keys to Carlo and does not reveal his own identity. The steps of retrieving transactions are as following:

1) Bob computes $R^* = k_{b1}R$ and $K^* = K_{b2} + K_{b3}$, and sends the set (R^*, K^*) anonymously to Carlo with a sum of agency fee.

2) Carlo computes $Y^* = H(R^*)G + K^*$, and retrieve whether there is a transaction on the blockchain that satisfies $Y^* = Y$. If yes, it stands for this transaction belonging to Bob, and Carlo issues an announcement about this transaction.

3) Once Bob observes this announcement, he looks for this transaction on the blockchain.

4) Bob computes again $H(k_{b1}R)G + K_{b2} + K_{b3}$ to ensure this transaction belong to him.

4.5 Spend Protocol

Bob takes advantage of his own three private keys for computing $x = H(k_{b1}R) + k_{b2} + k_{b3}$, and x is exactly the private key of the intermediate address. When Bob wants to spend the funds, he uses x for signing it. Because only Bob has the private key corresponding to the intermediate address, he can spend the funds instead of others.

5 Analysis

5.1 Anonymity

There are three participants as Alice, Bob and Carlo in our scheme. Carlo is only an agency, who has no knowledge about the relationship between Alice and Bob. Alice as sender signs the transaction by utilizing the linkable ring signature algorithm based on ECC. The verifier can confirm that the signature is generated by someone included in the group. The real identity of signer is absolutely anonymous for any verifier. Therefore, the linkable ring signature ensures the anonymity of the sender.

The intermediate address is used for concealing the real identity of receiver. Alice uses a part of Bob's public keys, and generates the intermediate address. Alice pays the funds to the intermediate address instead of the real address of Bob. No one has the ability to judge who the intermediate address belongs to, except Alice and Bob. Therefore, the intermediate address ensures the anonymity of the receiver. Although we utilize the agency Carlo to help receiver Bob retrieve transactions when he has finite computing resources, Carlo has no knowledge about the real identity of Bob. Bob sends a message anonymously to Carlo, and Carlo replies with an announcement. The receiver Bob is still anonymous for agency Carlo.

5.2 Resistant to Double-spending

Our scheme not only achieves completely anonymous transaction, but also can resist double-spending. When

users spend a sum of money, he must sign it by utilizing linkable ring signature in our scheme. Due to the linkability of linkable ring signature, nobody can spend the same money twice. We make use of the linkable tag to realize the linkability and resist double-spending.

Especially, the linkable tag is generated by utilizing the private key k_j of signer and the signature μ of previous transaction on the blockchain. With emphasis, μ stands for the source of the funds in the current transaction and it is public and unique. In addition, every user has unique private key. Therefore, if the malicious user attempts to spend the same money twice, he must sign it twice and result in generating two signatures with the same linkable tags. When there are the same linkable tags, the verifier can conclude that the user spends the same money twice, that is to say, double-spending. Once the verifier finds double-spending, he can reject directly the transaction. Therefore, the malicious user can certainly fail to double-spending.

5.3 Resistant to Denial of Receiver

Since the intermediate address conceals the real identity of the receiver, the receiver can deny that he receives the payment from sender. To solve this problem, the payment evidence R plays a important role in our scheme. R is recorded in the transaction and public to all nodes on the blockchain, but only the sender knows r. When the receiver deny that he receives the payment from sender, the sender can disclose r and prove that he pays indeed a sum of money to the intermediate address and it belongs to the receiver. Once the sender discloses r, all nodes on the blockchain can verify the correctness of $R = rG$. Therefore, all nodes on the blockchain can prove the innocence of sender and expose the denial behavior of receiver.

6 Conclusions

It is transaction information that are public on the blockchain leading to leakage about privacy information. Our scheme aims to ensure anonymous transaction and preserve the identity privacy of two sides in the transaction. We make use of the linkable ring signature to conceal the real identity of sender, and the intermediate address to conceal the real identity of receiver. The funds are deposited in the intermediate address, and only the receiver can spend it, so users do not have to worry about the funds being stolen. Nobody has ability to know the relation of sender and receiver. There is a agency, but he has no knowledge of two sides in the transaction. Furthermore, we add a agency in our scheme only when the receiver has limited computing resources. Certainly, the receiver can also retrieve transactions by himself if he has plenty of computing resources. In short, our scheme achieves completely anonymous. All algorithms used in our scheme are based on ECC having high security, and fully compatible with Bitcoin. It is worth noting that users have three public and private key pairs in our scheme, resulting in a large amount of computation. However, It is significant for us to ensure completely anonymous transactions at the expense of a large amount of computation.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. U143310218), by the Chunhui Project of Education Ministry of China (No. Z2014045), by the Science and Technology Bureau Project of Chengdu Municipality (No. 2016-XT00-00015-GX).

References

[1] G. D. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore, "Sybil-resistant mixing for bitcoin," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pp. 149–158, 2014.

[2] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: anonymity for bitcoin with accountable mixes," in *International Conference on Financial Cryptography and Data Security*, pp. 486–504, 2014.

[3] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[4] K. Gu, L. Y. Wang, N. Wu, and N. D. Liao, "Traceable certificateless ring signature scheme for no full anonymous applications," *International Journal of Network Security*, vol. 20, no. 4, pp. 762–773, 2018.

[5] E. Heilman, F. Baldimtsi, and S. Goldberg, "Blindly signed contracts: anonymous on-blockchain and off-blockchain bitcoin transactions," in *International Conference on Financial Cryptography and Data Security*, pp. 43–60, 2016.

[6] M. S. Hwang, S. F. Tzeng, and C. S. Tsai, "Generalization of proxy signature based on elliptic curves," *Computer Standards and Interfaces*, vol. 26, no. 2, pp. 73–84, 2004.

[7] M. H. Ibrahim, "Securecoin: a robust secure and efficient protocol for anonymous bitcoin ecosystem," *International Journal of Network Security*, vol. 19, no. 2, pp. 295–312, 2017.

[8] A. Judmayer, N. Stifter, K. Krombholz, and E. Weippl, "Blocks and chains: introduction to bitcoin, cryptocurrencies, and their consensus mechanisms," *Synthesis Lectures on Information Security Privacy and trust*, vol. 9, no. 1, pp. 1–123, 2017.

[9] A. Kosba, A.Miller, E. Shi, Z. K. Wen, and C. Papamanthou, "Hawk: the blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE symposium on security and privacy*, pp. 839–858, 2016.

[10] I. C. Lin and T. C. Liao, "A survey of blockchain security issues and challenges," *International Journal of Network Security*, vol. 19, no. 5, pp. 653–659, 2017.

[11] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," in *Australasian Conference on Information Security and Privacy*, pp. 325–335, 2004.

[12] L. H. Liu, Z. Z. Guo, Z. J. cao, and Z. Chen, "An improvement of one anonymous identity-based encryption scheme," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, pp. 11–21, 2018.

[13] Y. Liu, X. T. Liu, C. J. Tang, J. Wang, and L. Zhang, "Unlinkable coin mixing scheme for transaction privacy enhancement of bitcoin," *IEEE Access*, vol. 6, pp. 23261–23270, 2018.

[14] G. Maxwell, "Coinjoin: bitcoin privacy for the real world," in *Post on Bitcoin forum*, 2013.

[15] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*, pp. 397–411, 2013.

[16] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Consulted*, 2008.

[17] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565, 2001.

[18] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: practical decentralized coin mixing for bitcoin," in *European Symposium on Research in Computer Security*, pp. 345–364, 2014.

[19] N. V. Saberhagen. "Cryptonote v 2.0,", 2013.

[20] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, pp. 459–474, 2014.

[21] J. H. Seo, T. Kobayashi, M. Ohkubo, and K. Suzuki, "Anonymous hierarchical identity-based encryption with constant size ciphertexts," in *Proceedings of public key cryptography (PKC' 09)*, pp. 215–234, Irvine, CA, USA, March 2009.

[22] Q. C. ShenTu and J. P. Yu, "A blind-mixing scheme for bitcoin based on an elliptic curve cryptography blind digital signature algorithm," *Computer Science*, 2015.

[23] 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.

[24] L. Valenta and B. Rowan, "Blindcoin: blinded, accountable mixes for bitcoin," in *International Conference on Financial Cryptography and Data Security*, pp. 112–126, 2015.

[25] C. C. Yang, T. Y. Chang, and M. S. Hwang, "A new anonymous conference key distribution system based on the elliptic curve discrete logarithm problem," *Computer Standards and Interfaces*, vol. 25, no. 2, pp. 141–145, 2003.

[26] C. Yuan, M. X. Xu, and X. M. Si, "Research on a new signature scheme on blockchain," *Security and Communication Networks*, vol. 2017, no. 2, pp. 1–10, 2017.

[27] J. H. Ziegeldorf, R. Matzutt, M. Henze, F. Grossmann, and K Wehrle, "Secure and anonymous decentralized bitcoin mixing," *Future Generation Computer Systems*, vol. 80, pp. 448–466, 2018.

Biography

Yang Liu is a Master Candidate at the School of Computer and Software Engineering, Xihua University. Her research interests include Cryptography and blockchain technology.

Mingxing He is a Professor at the School of Computer and Software Engineering, Xihua University. He is the member of the International Association for Cryptologic Research, and is also the member of the CCF and ACM. His research interests include Cryptography and Network Security.

Fangyuan Pu is a Master Candidate at the School of Computer and Software Engineering, Xihua University. Her research interests include Network Security and Security Multi-Party Computation.

Malware Traffic Classification Based on Recurrence Quantification Analysis

Zheng-Zhi Tang[1,2], Xue-Wen Zeng[1], Zhi-Chuan Guo[1], and Man-Gu Song[1]
(Corresponding author: Zhi-Chuan Guo)
National Network New Media Engineering Research Center & Institute of Acoustics, Chinese Academy of Sciences[1]
Beijing 100190, China
(Email: guozc@dsp.ac.cn)
University of Chinese Academy of Sciences, Beijing 100049, China[2]
(Received Sept. 17, 2018; Revised and Accepted Mar. 2, 2019; First Online July 16, 2019)

Abstract

To characterize the behavioral characteristics of different malware traffic more intuitively and identify malware traffic more accurately, a novel analysis and identification method based on recurrence property of malware traffic is proposed. According to the real malware traffic sequences generated by different malwares, a high-dimensional phase space of the malware traffic sequences is constructed, and then the recurrence properties of the state trajectories of malware traffic are analyzed to reveal their inherent behaviors. By analyzing feature vector acquired by Recurrence Quantification Analysis (RQA) statistically and being combined with machine learning, malware traffic can be well identified. Comparing with the traditional method which uses the common flow statistical features, the proposed method has higher classification accuracy (about 96.55%) using fewer features.

Keywords: Recurrence Plots; Recurrence Quantification Analysis; Recurrence Property; Malware Traffic

1 Introduction

The rapid growth of network traffic has enriched the Internet content, while the network security issue has become increasingly prominent. Viruses, Trojans, worms and other malicious software hidden in the network not only effect the service quality of Internet Service Providers (ISP), but also pose great challenges in the field of data security and privacy protection of Internet users for the cloud computing [1,2] or cloud storage service [10,14], and even threaten national security. Therefore, the detection and classification of malware behavior has become the focus of current researchers.

At present, the detection of malware behavior mainly focuses on the detection of behavioral characteristics of malware itself [4], but the general malware itself has strong concealment and can hardly detect. Therefore, it is possible to detect and analyze the malware traffic behavior. The most important purpose of the analysis for network traffic behavior is to detect and discover some abnormal behavior of network traffic. Currently, the data attributes used to detect abnormal behavior are mainly statistical measurements of network traffic at different composition sizes. The detection methods for extracting these data attributes can be categorized into two main categories as follows [6]:

1) The dimension values of the network packet header are taken as data attributes directly, such as source/destination IP, source/destination port, protocol type, packet length and time of the packet;

2) The statistical characteristics of network traffic are used as data attributes, such as the traffic bytes between two hosts in a fixed time, the number of packets, the number of flows, and traffic entropy.

This paper discusses the detection and identification of malware from the perspective of traffic classification, and classifies the traffic generated by malware during network communication to identify malware traffic. In work of [21], the authors used the first 784 bytes of each session to form a 28*28 image, and then combined the convolutional neural network classifier to classify the malware traffic. In work of [13], the authors presented a novel malware classification method based on clustering of flow features and sequence alignment algorithms for computing sequence similarity, which represents network behavior of malware. However, the flow features used by authors include the IP address and port number, which are not rigorous. In work of [20], the authors used deep learning techniques to malware classification by their binary files. In work of [23], the authors demonstrated how ELIDe identifies malware within network traffic based on partially trained malware signature patterns that have significant weighted values within the classifier's weight vector. In work of [8, 19], the authors used common flow statistical features to classify network protocols or applications and achieve good results. But they did not in-

volve malware traffic. Currently, some researchers point out that the Internet is a complex network system, and its traffic behavior has nonlinear, non-stationary and other chaotic characteristics [3, 7]. In work of [24], the authors applied non-linear theory to analyze the traffic behavior of normal network applications, revealing the inherent characteristics of network behavior for different normal network applications. But nearly there is no research on the application classification issue by using recurrence property. In this paper, we use non-linear theory to analyze the inherent characteristics of malware traffic behavior, and use recurrence quantification analysis to extract the features of normal application traffic and malware traffic. Then it is combined with machine learning to classify and identify.

The main contributions of this paper are as follows:

- We propose a flow feature extraction method based on recurrence quantification analysis for malware traffic or normal traffic classification;
- For malware traffic or normal traffic, we firstly obtain TCP or UDP flows according to the five-tuple. For TCP or UDP flows, we obtain fixed-length sequences of packet size. Then we extract feature vectors by using recurrence quantification analysis on these sequences. Finally, the feature vectors are used as input of machine learning to classify;
- We directly apply common flow statistical feature based classification methods to the malware traffic classification. We extract flow feature set from raw network capture by using open source Netmate tool.
- We carry out many experiments on the machine learning to evaluate the performance of flow feature extraction method proposed. We compared the flow features that we extracted with the flow features that are commonly used to traffic classification in term of classification accuracy. In the case of same feature number for two methods, the proposal outperforms 11.99% the common technique in term of classification accuracy.

The rest of this paper is organized as follows. In Section 2, we will elaborate on recurrence plots and recurrence quantification analysis. In Section 3, we give a detailed description of proposed flow feature extraction algorithm based on recurrence quantification analysis and establish an analytical framework combined with machine learning to evaluate its performance. In Section 4, we explain the experimental process and analyze the experimental results. Finally, Section 5 concludes the work and analyzes possible future studies.

2 Recurrence Plots and Recurrence Quantification Analysis

The recurrence plots analysis method was first proposed by Eckman *et al.* in 1987 [5]. It is an important method for visualizing the periodicity, chaos and non-stationarity of time series by recurrence analysis on phase space. At present, it is mainly used for qualitative analysis of nonlinear dynamic systems and suitable for short time series. It also can reveal the internal structure of time series and give prior knowledge about similarity and predictability.

2.1 Phase Space Reconstruction

For the time series of chaotic systems, both the calculation of chaotic invariants or the establishment and prediction of chaotic models are carried out in phase space. Therefore, a phase space reconstruction is a very important step in chaotic time series processing. The phase space reconstruction is to reconstruct the state motion trajectory of the phase space system of the original time series by mapping the one-dimensional time series to the high-dimensional phase space. There are two main methods for phase space reconstruction: derivative reconstruction and coordinate delay reconstruction, which were proposed by Packard *et al.* in 1980 [18]. In the study of chaotic time series, the phase space reconstruction method of coordinate delay is widely used. Assuming an original one-dimensional time series $\{x_1, x_2, ..., x_n\}$, then each row vector of the m-dimensional phase space vector obtained by the phase space reconstruction is:

$$\mathbf{X}_i = \{x_i, x_{i+\tau}, ..., x_{i+(m-1)\tau}\} \quad (1)$$

where $i = 1, 2, ..., n-(m-1)\tau$ and τ is the delay time. It can be seen from Equation (1) that the choice of two parameters, embedding dimension m and delay time τ, is crucial for phase space reconstruction. Only by properly selecting the embedding dimension and delay time can the characteristics of the original system be accurately characterized. At present, there are many calculation methods for embedding dimension and delay time. We adopt the method that is most commonly used by researchers. We use false nearest neighbors (FNN) for the calculation of embedding dimension and the calculation of delay time uses mutual information (MI) [12].

2.2 Recurrence Plots

The recurrence phenomenon represents the recurrence of the phase space trajectory to a certain state, which is a fundamental property of deterministic dynamical systems, that is, the evolutionary pattern of the system state motion trajectory appears periodic recursive phenomenon [24]. According to the recurrence phenomenon, Eckmann *et al.* proposed the concept of recurrence plots [5]. Through the method of recurrence plots analysis, the motion trajectory of the phase points in the high-dimensional phase space can be visually represented in two-dimensional space.

The recurrence plots consist of white and black points in a two-dimensional square matrix. The white dots in the two-dimensional square matrix indicate that the two phase points are far away, and the black dots indicate

that the two phase points are close. The mathematical expression of the recurrence plots is:

$$R_{i,j} = \Theta(\epsilon - ||\mathbf{X}_i - \mathbf{X}_j||), i, j = 1, 2, ..., n - (m-1)\tau \quad (2)$$

where $R_{i,j}$ is a recurrence matrix element, when $R_{i,j} = 0$, it is represented as a white point on the recurrence plots, and when the value is 1, it is represented as a black point on the recurrence plots. $\Theta(x)$ is a Heaviside function, its value is 1 when the variable is greater than or equal to 0, and 0 when the variable is less than 0. $||x||$ is the Euclidean norm of the vector. ϵ is a pre-set threshold distance, and the choice of ϵ is critical for calculating recurrence plots. If its value is too large, the number of black points in the recurrence plots will be large, and its value is too small, which makes the white area in the recurrence plots large. In this paper, we choose 10% of the maximum diameter of the phase space as ϵ value, which called rule of thumb [16].

2.3 Recurrence Quantification Analysis

The recurrence plots are only qualitative and intuitive to show the recurrence property of the state motion trajectory of nonlinear systems. In the research, it is more desirable to quantitatively analyze. Recurrence Quantification Analysis (RQA) quantifies the characterization of recurrence plots by quantitative parameters, which are proposed by Zbilut *et al.* [22]. In this paper, the six typical RQA features, namely, recurrence rate (RR), determinism (DET), linemax (LMAX), entropy (ENT), laminarity (LAM), and trapping time (TT) are extracted to characterize the malware traffic or benign traffic. These features are used to characterize and describe the intrinsic characteristics of different malware traffic or benign traffic behaviors. Classification and identification are then performed based on the differences between these features. Herein, we give the definitions for RR, DET, and ENT. The detailed definitions for LMAX, LAM, and TT were stated in [15].

The recurrence rate (RR) represents the ratio of the recurrence point number to the entire phase point number, reflecting the density of the recurrence points. The recurrence rate is proportional to the periodicity of the time series. The formula is as follows:

$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} R_{i,j} \quad (3)$$

where N is the number of phase points and $R_{i,j}$ is the recurrence matrix element. The recurrence rate characterizes the recurrence degree of the system.

The determinism (DET) represents the ratio of the number of recurrence points which parallel to the main diagonal line segment to the number of total recurrence points in the recurrence plots. The determinism is positively correlated with the periodicity and predictability of the time series. The formula is as follows:

$$DET = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{i,j=1}^{N} R_{i,j}} \quad (4)$$

where l_{min} is the minimum diagonal segment length (generally 2) and $P(l)$ is the frequency of line segments of length l that parallel to the main diagonal. The determinism can be used to quantify the certainty of the system.

Entropy (ENT) represents the Shannon entropy of the 45° diagonal length probability distribution in a recurrence plots. The formula is as follows:

$$ENT = -\sum_{l=l_{min}}^{N} P(l) \ln P(l) \quad (5)$$

where $P(l)$ is the distribution probability of the main diagonal segment with length l, and l_{min} is the initial value of the length in the diagonal structure (generally 2). Entropy can be used to indicate the complexity of system certainty.

3 Classification Method Based on Machine Learning

The Gradient Boosting Decision Tree (GBDT) algorithm is known to improve the performance of a single classifier by combining several base classifiers that outperform every independent one. Now, it performs well in various data mining and machine learning methods. Furthermore, GBDT also performs well among solution methods for class imbalance problems [9]. In this part, we firstly propose a flow feature extraction algorithm based on recurrence quantification analysis. Then we propose the malware traffic classification method combined with GBDT.

3.1 Feature Extraction Algorithm

Algorithm 1 describes the complete flow feature extraction algorithm based on recurrence quantification analysis. The raw packets are processed to obtain the TCP or UDP flows. A flow is defined as all packets that have the same 5-tuple, i.e. source IP, source port, destination IP, destination port and transport protocol. Then all the flows of a normal or malicious application are combined into a PCAP file in order of timestamps, and the obtained PCAP file is processed to obtain a sequence Q of packet sizes. According to the length of initialized RQA sequence is l, the algorithm intercepts RQA sequence samples to extract features from sequence Q in order. Finally, the RQA method is used to obtain the feature set for the sequence Q.

3.2 Classification Method Process

Figure 1 shows the method of malware traffic classification process combined with GBDT. As shown in Figure 1,

Algorithm 1 Feature extraction

Begin
Initialize the *RQA sequence* length l.
Input raw packets and obtain UDP or TCP flows.
Combine all packets in flows in order of timestamps.
Obtain packet size sequence Q from ordered packets.
$L \Leftarrow$ Get sequence Q length
$n \Leftarrow \lfloor \frac{L}{l} \rfloor$
while $n > 0$ **do**
  Intercept *RQA sequence* from sequence Q in order
  Feature $\Leftarrow$ RQA(*RQA sequence*)
  $n \Leftarrow n - 1$
end while
return *Feature*
End

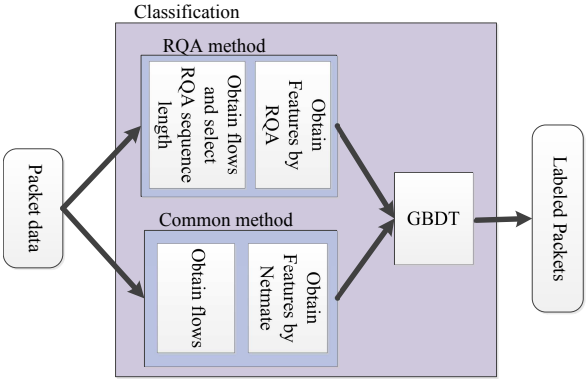


Figure 1: The method of malware traffic classification process combined with GBDT

the raw packets input can be selected by RQA method or commonly used method for feature extraction. Herein, the common feature extraction method is used for comparison experiment. As seen from Figure 1, the raw packets will be processed to obtain the flows, and the length of RQA sequence will be initialized in RQA method. Then the RQA method or the Netmate tool is used to extract the corresponding features respectively. Finally, the extracted features are used as input of the trained machine learning GBDT model for classification. Each of feature vectors input identifies a corresponding category label. In all experiments, we use the GBDT algorithm in the Scikit-learn for multi-classification.

4 Experiment and Result

To demonstrate the advantage of proposed flow feature extraction method based on the recurrence quantification analysis. We compare the flow features that we extracted by RQA with the common flow statistical features which are representative in traffic classification in term of classification performance by experiments.

4.1 Description of Dataset

The dataset used in the experiments is randomly selected from the CTU dataset, USTC-TFC2016 dataset [21] and VPN-nonVPN dataset [8]. A total of 10 normal application traffic and malware traffic capture are randomly selected here. The dataset is shown in Table 1.

Table 1: Malware traffic and benign traffic dataset

Malware traffic		**Benign traffic**	
Name	*Source*	*Name*	*Source*
Zeus	CTU	Hangouts	VPN-nonVPN
Miuref	CTU	HTTPS	CTU
Trickbot	CTU	P2P	CTU
Sennoma	CTU	SFTP	VPN-nonVPN
Artemis	CTU	SMB	USTC-TFC2016

4.2 Experiment Setup and Evaluation Metrics

The experimental platform is DELL R720 server which is equipped with CentOS release 7.3 operate system. The CPU is a 16-cores XeonE5620 2.40 GHz, and the memory is 16 GB. In all experiments, the classifier is GBDT algorithm and we carry out a grid search on parameter space to achieve the best classification accuracy with GBDT parameters are *random_state*=10, *n_estimators*=400, *max_depth*=6. In this paper, four evaluation metrics are used: accuracy (A), precision (P), recall (R), f1 value (F1). Accuracy is used to evaluate the overall performance of a classifier. Precision, recall and f1 value are used to evaluate performance of every class of traffic.

$$A = \frac{TP + TN}{TP + FP + FN + TN} \quad P = \frac{TP}{TP + FP}$$
$$R = \frac{TP}{TP + FN} \quad F_1 = \frac{2PR}{P + R} \tag{6}$$

where TP is the number of instances correctly classified as X, TN is the number of instances correctly classified as Not-X, FP is the number of instances incorrectly classified as X, and FN is the number of instances incorrectly classified as Not-X.

4.3 Common Flow Statistical Features for Classification

In the open literature, Moore *et al.* presented one of the earliest results about classification of network flows into protocol categories by using flow features combined with supervised machine learning [17]. The authors studied discriminators, attributes primarily derived from network flows and the feature set consists of 248 statistical characteristics. Follow-up research in this area mainly focused

on the selection of flow feature sets and the machine learning methods. In this paper, we extract the common flow statistical feature sets from raw network capture by using open source Netmate tool. The extracted feature set consists of 44 common flow statistical features. We only use 40 common flow statistical features that remove the IP address and port number. A detailed description of the common flow statistical features extracted by Netmate can be found on the Netmate official website.

Herein, the Netmate tool is used to process the data set in Table 1, and the number of flow feature samples of each normal or malicious application is shown in Table 2. In order to reduce the impact of class imbalance on the classification results, the data is under-sampled for the class with a large number of samples, and the SMOTE method is applied to the class with a small number of samples. The final preprocessed result is as shown in Table 2.

As seen in Table 3, for the class imbalance samples, the classification accuracy after samples undersampling is higher than that processed by the undersampling and SMOTE combination method. The classification accuracy after samples undersampling outperforms 1.9%. However, undersampling only reduces the number of classes with a large number of samples and the number of classes with a small number of samples is still small. So the class imbalance is still obvious. This can be seen from Table 2. As seen in Table 4, due to the SFTP sample is the least, the recall of SFTP is only 50% and F1-score is only 67%. The SFTP identification result is very poor. The overall classification result is good by using the combination of undersampling and SMOTE. Except for the precision mean, the recall mean and the F1-score mean are higher. In view of the importance for the small class identification and the suggestion that if the training sample size is too large, a combination of SMOTE and undersampling is an alternative [9]. Finally, a comprehensive consideration is given to the use of samples processed by a combination of under-sampling and SMOTE in follow-up experiments.

4.4 Recurrence Quantification Analysis Based Flow Features for Classification

In this section, the flow features extracted by recurrence quantification analysis are performed through experiments. The reason for selecting the sequence of packet sizes is that the continuation of packet size on the timeline can well show the inherent characteristics of network behavior, such as periodicity, data transmission characteristics and so on for the malware traffic or benign traffic. The difference of network behavior characteristics will also lead to difference in the characteristics of the non-linear dynamic system of network traffic. Therefore, the recurrence analysis for the sequence of malware traffic packet size or benign traffic packet size can reveal its unique network behavior characteristics. They can be classified by machine learning methods based on their unique characteristics.

Herein, the raw packets are processed to obtain samples by Algorithm 1 in Section 3.1. The sequence length of each normal or malicious application is obtained by random sampling method as shown in the Table 5. Then, a subsequence of length n (n = 40, 60, 80, 100 in this paper) is taken as one sample, and finally the number of samples of each normal or malicious application in the case of sub-sequences with different lengths is shown in Table 5.

4.4.1 Embedding Dimension and Delay Time

In Section 2.1, it is mentioned that the false nearest neighbor method and mutual information method are used to calculate the embedding dimension and delay time respectively. According to the principle of embedding dimension is determined by the false nearest neighbor method in [12], by increasing the size of embedding dimension one by one, and then calculating the proportion of adjacent errors under each embedding dimension, the first embedding dimension which makes the proportion close to 0 (less than 0.05) or the proportion of adjacent errors no more reduce is the best embedding dimension. The calculation formula for the adjacent error is as follows:

$$r_i = \frac{||\mathbf{X}_{j+1} - \mathbf{X}_{i+1}||}{||\mathbf{X}_j - \mathbf{X}_i||} \tag{7}$$

where $\mathbf{X}_i$,$\mathbf{X}_j$ is the phase point in the m-dimensional phase space, $\mathbf{X}_{i+1}$,$\mathbf{X}_{j+1}$ is the phase point in the $m+1$-dimensional phase space, r_i is the ratio of the distance from the phase point $\mathbf{X}_{j+1}$ to the phase point $\mathbf{X}_{i+1}$ and the distance from the phase point $\mathbf{X}_j$ to the phase point $\mathbf{X}_i$, if r_i is greater than the determined threshold r (generally 2), then the phase point $\mathbf{X}_i$ and the phase point $\mathbf{X}_j$ are adjacent errors.

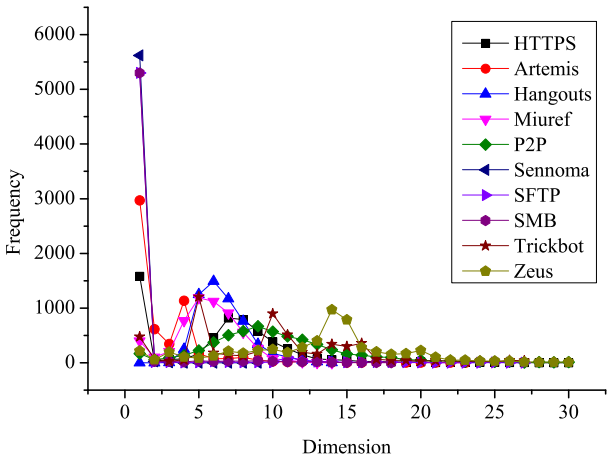


Figure 2: Frequency of the embedding dimension for subsequence samples with length 80

Figure 2 shows the frequency of embedding dimension distribution for subsequence samples with length 80. The

Table 2: Number of flow feature samples for malware traffic and benign traffic

Name	Origin	Undersampling	Undersampling+SMOTE
Zeus	227236	5681	5681
Miuref	4837	4837	4837
Artemis	221758	5687	5687
P2P	2209	2209	4418
HTTPS	6573	5651	5651
SFTP	24	24	5760
Sennoma	653	653	5877
SMB	214	214	5564
Trickbot	94515	5628	5628
Hangouts	1357	1357	5428

Table 3: Classification accuracy of class imbalance after different processing

Method	Undersampling	Undersampling+SMOTE
Accuracy	0.9678	0.9488

Table 4: Precision, recall, F1-score for class imbalance after different processing

Method	Undersampling			Undersampling+SMOTE		
Evaluation	*Precision*	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*
Miuref	1.00	1.00	1.00	1.00	1.00	1.00
SMB	1.00	1.00	1.00	1.00	1.00	1.00
Zeus	0.97	0.97	0.97	0.97	0.95	0.96
Trickbot	1.00	1.00	1.00	1.00	1.00	1.00
P2P	0.86	0.83	0.84	0.85	0.82	0.84
Hangouts	0.97	0.96	0.96	0.99	0.78	0.87
SFTP	1.0	**0.5**	**0.67**	0.99	0.97	0.98
HTTPS	0.93	0.94	0.93	0.94	0.95	0.95
Artemis	1.00	1.00	1.00	1.00	1.00	1.00
Sennoma	0.96	0.97	0.97	0.80	0.99	0.95
Average	0.969	0.917	0.934	0.954	0.946	0.955

Table 5: The number of samples of each normal or malicious application

Name	Length	40	60	80	100
Zeus	446400	11446	7566	5650	4509
Miuref	446700	11453	7571	5654	4512
Artemis	445700	11428	7554	5641	4502
P2P	445500	11423	7550	5639	4499
HTTPS	445100	11412	7544	5634	4495
SFTP	444700	11402	7537	5629	4491
Sennoma	446400	11446	7566	5650	4509
SMB	444900	11407	7540	5631	4493
Trickbot	446000	11435	7559	5645	4505
Hangouts	443900	11382	7523	5618	4483

final embedding dimension for each normal or malicious application is the embedding dimension with the largest distribution frequency. The final embedding dimension statistics are shown in Table 6. As seen from Table 6, the embedding dimension of each normal or malicious application remains basically unchanged in the case of subsequence samples with different lengths.

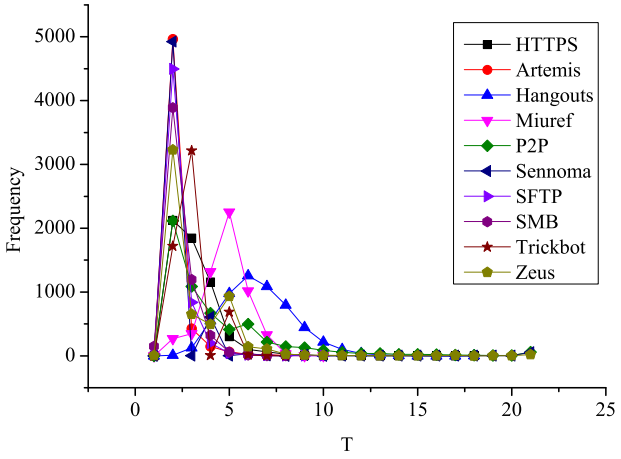


Figure 3: Frequency of the delay time for subsequence samples with length 80

The calculation formula for delay time by the mutual information method is as follows [12]:

$$S = -\sum_{i,j} P_{i,j}(t) \ln \frac{P_{i,j}(t)}{P_i P_j} \tag{8}$$

where P_i and P_j are the probabilities of the points falling into the segments i and segments j in the traffic sequences respectively, $P_{i,j}(t)$ is the probability that the two points with the interval time t fall into the segments i and segments j respectively. The mutual information under each delay time is calculated by the formula, and the delay time corresponding to the first mutual information with local minimum value is the optimal delay time τ.

Figure 3 shows the frequency of the delay time distribution for subsequence samples with length 80. The final delay time for each normal or malicious application is the delay time with the largest distribution frequency. The final delay time statistics are shown in Table 6. As seen from Table 6, the delay time of each normal or malicious application remains basically unchanged in the case of subsequence samples with different lengths.

According to the embedding dimension and delay time, the recurrence plots of each normal or malicious application can be calculated by Formula 2. Then the recurrence quantification analysis method is applied to each recurrence plots, and finally, RR, DET, LAM, ENT, LMAX, and TT are obtained to form the feature vectors of each normal or malicious application. The feature vectors of each normal or malicious application are used as input of GBDT to classify. Figure 4 is the classification accuracy for each normal or malicious application in the case of subsequence samples with length 40, 60, 80, 100, respectively. As shown in Figure 4, as the subsequence length increases, the classification accuracy also increases. When the length is 80, the classification accuracy reaches a maximum value 96.55% and then begins to decrease. This shows that when the subsequence length is 80, the inherent unique characteristics of each normal or malicious application can be well represented by recurrence quantification analysis. In follow-up experiments, we will use the subsequence samples with length 80.

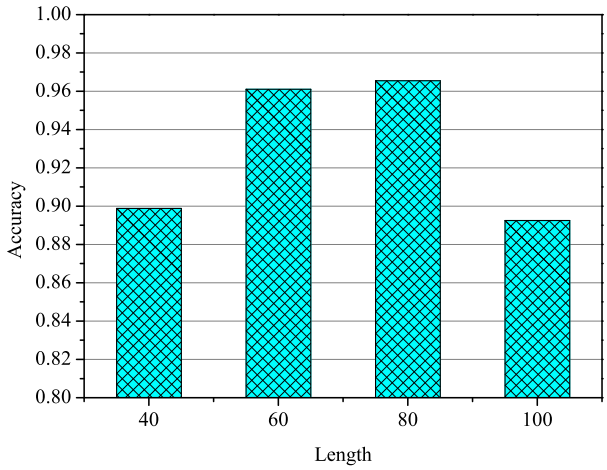


Figure 4: Classification accuracy of normal or malicious applications in the case of subsequences with different lengths

4.5 Comparison

In this section, we will compare the flow features that we extracted by recurrence quantification analysis with the flow statistical features that are commonly used to traffic classification in term of classification performance. Since there are only 6 kinds of flow features extracted by the proposed method, however, the type number of common flow statistical features extracted by the Netmate tool is 40. Herein, we carry out an experiment for choosing the number of common flow statistical features that can reach the best classification accuracy.

Figure 5 is a classification accuracy using different numbers of common flow statistical features randomly selected. It can be seen from the Figure 5 that as the number of random common flow statistical features increases, the classification accuracy increases, while the rising rate becomes slowly. When all the 40 features are used, the classification accuracy reaches the maximum value 94.53%. In follow-up experiments, we compare the 6 and all 40 common flow statistical features randomly selected with the 6 flow features extracted by recurrence quantification analysis in term of classification accuracy respectively.

Table 6: Embedding dimensions and delay time for different length subsequence samples of each normal or malicious application

Name	Parameter	40	60	80	100
HTTPS	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
Artemis	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
Hangouts	*Dimension*	5	5	6	7
	Delay time	5	5	6	6
Miuref	*Dimension*	1	5	5	7
	Delay time	5	5	5	5
P2P	*Dimension*	1	8	9	11
	Delay time	2	2	2	2
Sennoma	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
SFTP	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
SMB	*Dimension*	1	1	1	1
	Delay time	2	2	2	2
Trickbot	*Dimension*	5	5	5	5
	Delay time	3	3	3	3
Zeus	*Dimension*	1	14	14	15
	Delay time	2	2	2	2

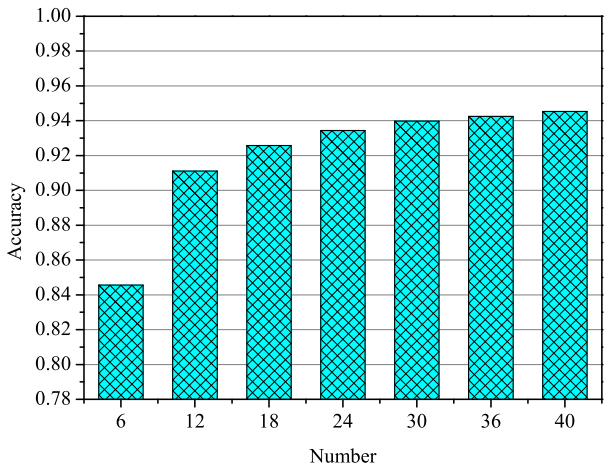


Figure 5: Classification accuracy corresponding to different numbers of common flow features randomly selected

Figure 6 shows the comparison of classification accuracy for the 6 and all 40 common flow statistical features randomly selected with the 6 flow features extracted by recurrence quantification analysis. The left column indicates the classification accuracy by using common flow statistical features, which is abbreviated as CFF for the convenience of description. The right column represents the classification accuracy of the 6 flow features extracted by recurrence quantification analysis. Also for the convenience of description, we abbreviate it as RQA. As shown in Figure 6, when the number of RQA and CFF features is 6, the classification accuracy of RQA is obviously better than CFF, and it outperforms 11.99%. When using all 40 common flow statistical features, the classification accuracy is much higher indeed, but the classification accuracy is still lower than that using 6 flow features extracted by the recurrence quantification analysis. The RQA outperforms 1.67%.

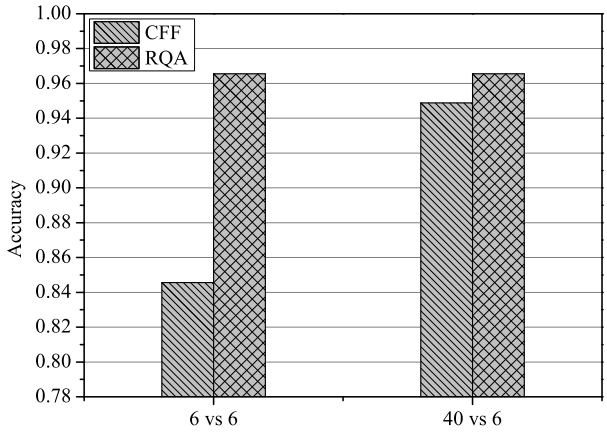


Figure 6: Comparison of classification accuracy between 6 flow features extracted by RQA and random 6 or 40 common flow statistical features

Table 7 shows the precision, recall, F1-score for 6 flow features extracted by RQA and all 40 common flow sta-

Table 7: Precision, recall, F1-score for 6 flow features extracted by RQA and all 40 common flow features

Method	**RQA**			**CFF**		
Evaluation	*Precision*	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*
Miuref	1.00	1.00	1.00	1.00	1.00	1.00
SMB	0.92	0.92	0.92	1.00	1.00	1.00
Zeus	0.98	0.98	0.98	0.97	0.95	0.96
Trickbot	0.99	1.00	0.99	1.00	1.00	1.00
P2P	0.97	0.98	0.98	**0.85**	**0.82**	**0.84**
Hangouts	0.99	0.99	0.99	0.99	**0.78**	**0.87**
SFTP	0.96	0.95	0.96	0.99	0.97	0.98
HTTPS	0.91	0.93	0.92	0.94	0.95	0.95
Artemis	0.94	0.93	0.93	1.00	1.00	1.00
Sennoma	1.00	1.00	1.00	**0.80**	0.99	0.95
Average	0.966	0.968	0.967	0.954	0.946	0.955

tistical features. As seen from Table 7, the precision mean, the recall mean, and the F1-score mean of the RQA method are better than the CFF method. When using the RQA method proposed, the classification precision, recall, and F1-score of each normal or malicious application are stable at more than 91%. However, when using the CFF method, some evaluation values of P2P, Sennoma, and Hangouts are significantly less than 90%. From this, it can be concluded that the RQA method proposed has obvious advantages over the CFF method.

4.6 Analysis and Discussion

From the above experiments, obviously, the proposed flow features extracted by RQA performs better than the common flow statistical features extracted by Netmate tool. The proposed RQA method not only has higher classification accuracy, but also has better accuracy mean, recall mean, and F1-score mean. Moreover, the classification precision, recall, and F1-score of each normal or malicious application are stable at more than 91%. The most important thing is that the RQA method proposed uses only 6 flow features, but it performs better than the CFF method by using 40 common flow statistical features. Since only 6 flow features are used, fewer features mean less time consumption for training and classification. Therefore, the proposed RQA method is efficient and possibilities for real-time online classification.

For the classification of malware traffic, in the latest work of [21], the authors used the first 784 bytes of each session to form a 28*28 image, and then combined the convolutional neural network classifier to classify the malware traffic. Finally, its classification accuracy can reach about 99%. In this paper, we do not take the work of [21] as a comparison, mainly because the proposed RQA method is not similar to the method of [21] in principle. In the early work of [13], the authors presented a novel malware classification method based on clustering of flow features and sequence alignment algorithms. However, the authors took into account the IP address and port number in the flow features, which is not rigorous enough. In order to make a more scientific and fair comparison, we chose the current common flow statistical feature based traffic classification method [8, 19]. We use it directly on the malware traffic classification and compare the classification results with the proposed RQA method.

5 Conclusions and Future Work

Malware detection is an active and hot research area in network security issue, which governs identification performance. Motivated by identifying malware through traffic generated by malware communication, we propose a novel flow feature extraction method based on recurrence quantification analysis for malware traffic or normal traffic classification. Our goal is to reduce the high time consumption due to excessive flow features in classification and improve classification performance. The key characteristic of flow feature extraction method based on recurrence quantification analysis is to extract feature vectors by using recurrence quantification analysis on these sequences of packet size. The raw packets are processed to obtain the TCP or UDP flows. Then all the packets of a normal or malicious application are combined into a large PCAP file in order of timestamps, and the obtained PCAP file is processed to obtain a sequence of packet sizes. Finally, the feature vectors extracted by RQA are used as input of machine learning to classify. The sensitivity of this algorithm against different situations is studied. Experiments on the machine learning to evaluate the performance of proposed flow feature extraction algorithm verify that it has fewer flow features but higher classification accuracy than that using the common flow statistical features.

In the future, we will increase the types of malware traffic and benign traffic, and implement experiments in real-time systems, such as real-time data collection and

analysis system [11], to evaluate the classification accuracy of the proposed flow extraction method based on RQA over a longer period.

Acknowledgments

This work was supported by The Next-Generation Broadband Wireless Mobile Communications Network National Science and Technology of Major Projects (No. 2017ZX03001019). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers.

References

[1] D. S. AbdElminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40–48, 2018.

[2] M. H. R. Al-Shaikhly, H. M. El-Bakry, and A. A. Saleh, "Cloud security using markov chain and genetic algorithm," *International Journal of Electronics and Information Engineering*, vol. 8, no. 2, pp. 96–106, 2018.

[3] N. Bigdeli and M. Haeri, "Time-series analysis of tcp/red computer networks, an empirical study," *Chaos Solitons and Fractals*, vol. 39, no. 2, pp. 784–800, 2009.

[4] Z. Chen, Q. Li, P. Zhang, and P. Feng, "Signature selection for kernel malware based on cluster analysis (in chinese)," *Journal of Electronics and Information Technology*, vol. 37, no. 12, pp. 2821–2829, 2015.

[5] J. P. Eckmann and D. Ruelle, "Fundamental limitations for estimating dimensions and lyapunov exponents in dynamical systems," *Physica D Nonlinear Phenomena*, vol. 56, no. 2-3, pp. 185–187, 1992.

[6] Y. Fu, H. Li, X. Wu, and J. Wang, "Detecting apt attacks: a survey from the perspective of big data analysis (in chinese)," *Journal on Communications*, vol. 36, no. 11, pp. 1–14, 2015.

[7] K. Fukuda, "Observations and possible causes of phase transition phenomena in internet traffic," *Ipsj Magazine*, vol. 45, pp. 603–609, 2004.

[8] D. G. Gerard, L. A. Habibi, M. M. S. IsIam, and G. Ali, "Characterization of encrypted and vpn traffic using time-related features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, pp. 404–414, 2016.

[9] H. Guo, Y. Li, S. Jennifer, M. Gu, Y. Huang, and B. Gong, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems With Applications*, vol. 73, pp. 220–239, 2017.

[10] M. S. Hwang, T. H. Sun, and C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service," *Journal of Circuits Systems and Computers*, vol. 26, no. 5, 2017.

[11] D. S. Jiang, L. Xue, W. X. Kai, and L. C. Mei, "Design of real-time data collection and analysis system based on spark streaming (in chinese)," *Network New Media*, vol. 6, no. 5, 2017.

[12] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*. New York: Cambridge University Press, 2004.

[13] H. Lim, Y. Yamaguchi, H. Shimada, and H. Takakura, "Malware classification method based on sequence of traffic flow," in *Proceedings of the 1st International Conference on Information Systems Security and Privacy (ICISSP 2015)*, pp. 230–237, France, 2015.

[14] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of attribute-based access control with user revocation in cloud data storage," *International Journal of Network Security*, vol. 18, no. 5, pp. 900–916, 2016.

[15] N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, and J. Kurths, "Recurrence plot based measures of complexity and their application to heart-rate-variability data," *Phys. Rev. E*, vol. 66, p. 026702, 2002.

[16] Thiel M Marwan N, Romano M C, "Recurrence plots for the analysis of complex systems," *Physics Reports*, vol. 438, no. 5, pp. 237–329, 2007.

[17] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2005)*, pp. 50–60, Banff, Alberta, Canada, 2005.

[18] N. H. Packard, J. P. Crutchfield, and J. D. Farmer, "Geometry from a time series," *Physical Review Letters*, vol. 45, no. 9, pp. 712–716, 1980.

[19] A. Pektas and T. Acarman, "Identification of application in encrypted traffic by using machine learning," in *Proceedings of the 5th International Conference on Man-Machine Interactions (ICMMI 2018)*, vol. 659, pp. 545–554, 2018.

[20] R.K. Rahul, T. Anjali, V. K. Menon, and K. P. Soman, "Deep learning for network flow analysis and malware classification," *Communications in Computer and Information Science*, vol. 746, pp. 226–235, 2017.

[21] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proceedings of the 31st International Conference on Information Networking (ICOIN 2017)*, pp. 712–717, Da Nang, Vietnam, January 2017.

[22] C. L. Webber and J. P. Zbilut, "Dynamic assessment of physiological system and state using recurrence plot strategies," *Appl Physiol*, vol. 76, pp. 965–973, 1994.

[23] K. F. Yu and R. E. Harang, "Machine learning in malware traffic classifications," in *Proceedings of IEEE Military Communications Conference (MILCOM 2017)*, pp. 6–10, 2017.

[24] J. Yuan, J. Wang, Q. Li, and X. Chen, "Recurrence based nonlinear analysis for network application traffic (in chinese)," *Journal of Tsinghua University (Science and Technology)*, vol. 54, no. 4, 2014.

Biography

Zheng-Zhi Tang a Ph.D. candidate in signal and information processing from National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences (IACAS) and University of Chinese Academy of Sciences. His research interests include Network Security, Information Safety and ML (Machine learning).

Xue-Wen Zeng received the B.Sc. degree from Shanghai Jiao Tong University, Shanghai, China, and the M.Sc. and Ph.D. degrees in signal and information processing from Institute of Acoustics, Chinese Academy of Sciences (IACAS), Beijing, China. He is currently working at National Network New Media Engineering Research Center, IACAS as a research professor. His research interests include network new media technology, media information security, multimedia communication, digital broadcasting and signal processing.

Zhi-Chuan Guo received the B.Sc. degree in optical technology and photoelectric instrument from Wuhan University, Wuhan, China, and the Ph.D. degree in electronic circuit and system from University of Science and Technology of China, Hefei, China. He is currently working at National Network New Media Engineering Research Center, IACAS as an associate research professor. His research interests include network new media technology and FPGA hardware acceleration technology.

Man-Gu Song received the B.Sc. degree in computer science and technology from the Tianjin University of Technology and Education, Tianjin, China, and the M.Sc degree in Electronics and Communication Engineering from the School of Microelectronics, Chinese Academy of Science, Beijing, China. She is currently working at National New Media Engineering Research Center, IACAS as a research assistant. Her current research interests include FPGA hardware accelerate technology and research on national secret algorithm.

Secure and Efficient Client-Side Data Deduplication with Public Auditing in Cloud Storage

Qianlong Dang, Hua Ma, Zhenhua Liu, and Ying Xie
(Corresponding author: Qianlong Dang)

School of Mathematics and Statistics, Xidian University
No.2, South Taibai Road, Xi'an Shaanxi 710071, P.R. China
(Email: xidianqldang@163.com)

(Received Nov. 06, 2018; Revised and Accepted Feb. 7, 2019; First Online June 14, 2019)

Abstract

In this paper, we propose a secure and efficient client-side data deduplication scheme with public auditing. In the process of deduplication, the proposed scheme improves the probability that the cloud server detects the missing blocks by eliminating the aggregated proofs structure. Meanwhile, we combine the oblivious pseudo-random function protocol with proxy re-encryption technology to implement key distribution without online data owners or the authorized party. Moreover, during public auditing, proxy re-signature technology is utilized to require only one auditing tag for each data block. For data deduplication, the proposed scheme is a zero-knowledge proof of knowledge assuming that the Discrete Logarithm (DL) problem is hard. In addition, the symmetric key can not be recovered by the cloud server or malicious users. And security analysis indicates that our scheme is secure against adaptive chosen-message attack under the Computational Diffie-Hellman (CDH) assumption during public auditing. Finally, the performance evaluation demonstrates that the proposed scheme is practical and efficient.

Keywords: Cloud Storage; Key Distribution; Proof of Ownership; Public Auditing; Secure Deduplication

1 Introduction

In cloud storage services, clients outsource data to a remote storage and access the data whenever they need the data. Recently, owing to its convenience, cloud storage services have become widespread, and it can provide resource-constrained users with convenient storage and computing services [5, 16, 21]. Although the cloud storage offers many advantage, it also brings a huge storage burden and some security challenges such as data integrity [23].

Since cloud storage service is increasingly used, a large amount of data is gathered into the cloud server. IDC predicts that the cloud data will reach 44ZB in 2020. A recent survey conducted by Microsoft [22] indicates that about 90% of data stored in the cloud are duplicated copies. Regarding storage efficiency, commercial cloud storage services, such as Dropbox, Wuala and Bitcasa, adopt deduplication technique to store one copy of each data and refer other duplicates to this stored copy. However, several security threats potentially exist during deduplication [1, 6, 14, 15, 24]. For instance, if a malicious user needs to gain access to the data that already exists in the cloud server, he can pass the verification by only owing the hash value of the data rather than the original data. It is obvious that the cloud server cannot distinguish whether user indeed possess the data only through matching its hash value. Therefore, how to convince the cloud server that the user indeed possesses original data becomes an important problem.

As a promising approach, message-locked encryption (MLE) [3] was used as client-side deduplication schemes [14, 15]. However, almost all of these schemes adopt deterministic encryption method and are vulnerable to brute force dictionary attacks. To solve this issue, some schemes encrypt the data with a randomly selected symmetric key, and the first uploader distribute the symmetric key to subsequent uploaders by adopting the proxy re-encryption technology. Unfortunately, all existing key distribution processes require the assistance of online data owners or the authorized party. In order to improve security and efficiency, it is essential to consider how can the deduplication scheme implement key distribution without the assistance of online data owners or the authorized party.

When clients use cloud storage services, they have hopes of guaranteeing the completeness of cloud storage data [4, 11, 13, 18, 30]. Accordingly, we need an efficient way to check the integrity of data in remote storage. Clients decide to authorize the task of auditing to a third party auditor (TPA), which enables that clients can efficiently perform integrity verifications even without the local copy

of data. However, these integrity auditing schemes rarely consider secure client-side data deduplication. Therefore, it is essential to combine secure client-side deduplication with integrity auditing.

In this paper, aiming at solving both storage efficiency and data integrity, we concentrate on how to design a secure and efficient client-side data deduplication scheme with public auditing. Inspired by a proof of ownership protocol [28] and a key distribution process [6, 28], we will propose an efficient client-side data deduplication and public auditing scheme which achieves a better trade-off between functionality and efficiency through improving Liu *et al.*'s auditing scheme [20]. Our main contributions can be summarized as follows.

- We utilize the aggregated proofs structure and zero-knowledge proof for proof of ownership, which improves the probability that the cloud server detects the missing blocks. Meanwhile, we prove that the proof of ownership scheme is sound, complete and zero-knowledge.
- The proposed scheme integrates the oblivious pseudo-random function (O-PRF) protocol with proxy re-encryption technology to implement key distribution. In the process of data deduplication, the proposed scheme does not require the assistance of online data owners or the authorized party. The process shows that the symmetric key of data can not be recovered by the cloud server or malicious users.
- By adopting proxy re-signature technology, subsequent uploaders can verify integrity of the cloud storage data in the proposed scheme. We prove that the proposed auditing scheme can guarantee the correctness and unforgeability. Finally, the performance evaluation demonstrates that the proposed scheme is practical and efficient.

The rest of this paper is organized as follows. A review about some related works is given in Section 2. Some preliminaries are presented in Section 3. Section 4 defines system and security model. The concrete construction of secure and efficient client-side deduplication scheme with public auditing is detailed in Section 5. Section 6 analyzes the security of our scheme. Section 7 presents the performance evaluation. Finally, we conclude the paper in Section 8.

2 Related Works

With the development of cloud computing, a rising number of enterprises and organizations choose to outsource their data to the cloud server. Then, the large amount of data is gathered in the cloud server, which will bring huge storage overhead to the cloud server. Therefore, client-side deduplication technology is introduced to solve data redundancy problems. During a client-side deduplication system, after receiving the hash value of data from the user, the cloud server checks whether the duplicate exists in cloud storage. Nonetheless, Halevi *et al.* [10] explained several security attacks that may occur in client-side deduplication systems. Moreover, Halevi *et al.* [9] proposed the concept of proof of ownership (PoW). The purpose of proof of ownership is to better verify that a client owns the entire data instead of owning partial data. Some scholars have proposed a variety of PoW schemes [8, 25–27]. However, when data ownership is verified, the existing schemes are based on the hash of the data rather than the original data. That is to say, the clients could be accepted by the cloud server as data owners with the hash of the data, even if they do not have original data. To address this issue, Yang *et al.* [28] verified the data ownership by the original data block. Nevertheless, this scheme has a low probability of detecting missing blocks.

The vast majority of client-side deduplication schemes [14, 15] adopted message-locked encryption technology to achieve data deduplication. However, these schemes are vulnerable to brute force dictionary attacks. Yang *et al.* [28] presented a scheme that the first uploader encrypts the data by a random symmetric key and utilizes the proxy re-encryption technology to distribute symmetric keys to subsequent uploaders. This scheme requires the online data owner to assist key distribution. In addition, Ding *et al.* [6] introduced an authorized party to complete key distribution in the client-side deduplication scheme.

When a client stores data in the cloud server, the user loses the right of managing the data. Consequently, it is very vital for users to check the integrity of cloud storage data in time. Recently, some scholars have proposed a variety of auditing schemes [7, 12, 29]. However, these schemes fail to achieve secure deduplication. Li *et al.* [17] proposed an integrity auditing scheme for encrypted deduplication storage, which introduced a third-party cluster to generate the same signature tag for duplicate data, but brought some data privacy issues. In order to resolve this problem, Liu *et al.* [20] used the MLE and proxy re-signature to actualize the deduplication of auditing tags among users, which can protect data privacy and generate one tag for the identical data block. However, due to the adoption of message-locked encryption technology, this scheme is vulnerable to brute force dictionary attacks. In addition, this scheme adopts server-side deduplication, which causes huge network bandwidth consumption.

3 Preliminaries

We now explain some preliminary notions that will form the foundations of our scheme.

3.1 Bilinear Pairings

Let G_1 and G_T be two multiplicative cyclic groups of the same prime order q. Let $e : G_1 \times G_1 \to G_T$ denote a bilinear map [28] constructed with the following properties:

1) Bilinearity: For all $a, b \in Z_q^*$ and $g_1, g_2 \in G_1$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
2) Non-degeneracy: There exists a point g_1 such that $e(g_1, g_1) \neq 1$.
3) Computability: $e(g_1, g_2)$ for any $g_1, g_2 \in G_1$ can be computed efficiently.

3.2 Complexity Assumptions

Definition 1. *(Discrete Logarithm (DL) problem [6]) Given $g \in G_1$ and $y = g^x$, where x are selected uniformly at random from Z_q^*, it is hard to get x.*

Definition 2. *(Computational Diffie-Hellman (CDH) problem [6]) Given a group G_1 with generator g and elements $g^x, g^y \in G_1$, where x, y are selected uniformly at random from Z_q^*, it is hard to compute the value of g^{xy}.*

3.3 Oblivious Pseudo-Random Function Protocol

Oblivious pseudo-random function protocol (O-PRF protocol) is introduced in DupLESS [2]. This protocol generates secret value by interacting between the key server and the user instead of deriving the secret value from the data directly.

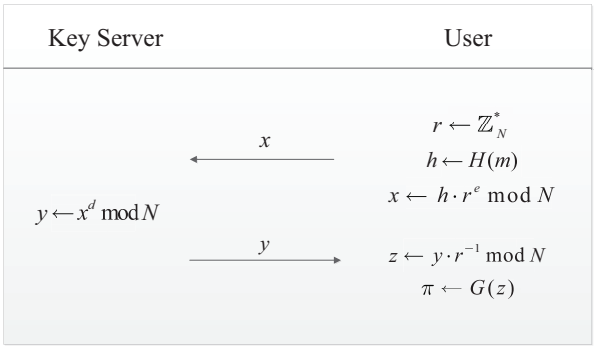


Figure 1: O-PRF protocol

Figure 1 illustrates the O-PRF protocol based on RSA blind signatures. The key server has secret key d and public key e where $ed \equiv 1 \bmod \Phi(N)$. $H : \{0,1\}^* \rightarrow Z_N^*$ and $G : Z_N^* \rightarrow \{0,1\}^*$ are two secure hash functions. The interaction process is as follows. (1) The uploader calculates the hash value $h = H(m)$ of the data and selects a random value $r \in Z_N^*$. Moreover, the uploader computes the blinded hash $x = h \cdot r^e \bmod N$ and sends x to the key server. (2) Upon receiving x, the key server computes $y = x^d \bmod N$ and sends y to the uploader. (3) The uploader calculates $y \cdot r^{-1} \bmod N$ and obtains secret value z. Finally, the uploader computes the secret value $\pi \leftarrow G(z)$.

The cloud server does not have the hash value h of data, so it cannot generate the secret value π. Moreover, the secret value generation process will not disclose any information. A malicious user who does not have a secret key d, therefore, cannot generate secret value π. It allows encryption to be secure against the brute force attacks even for predictable message set.

3.4 Aggregated Proofs Structure

During the verification process, the aggregated proofs [19] can improve verification efficiency and save network bandwidth. By using the idea of aggregated proofs, we design the aggregated proofs structure (As shown in Figure 2). The first uploader uploads the original proofs to the cloud server. Firstly, the original proofs are multiplied by the selected coefficient to obtain the first-level proofs. Secondly, the second-level proofs are generated by multiplying two adjacent terms in the first-level proofs. Thirdly, the third-level proofs are generated by multiplying two adjacent terms in the second-level proofs. Finally, the j-level proofs are calculated.

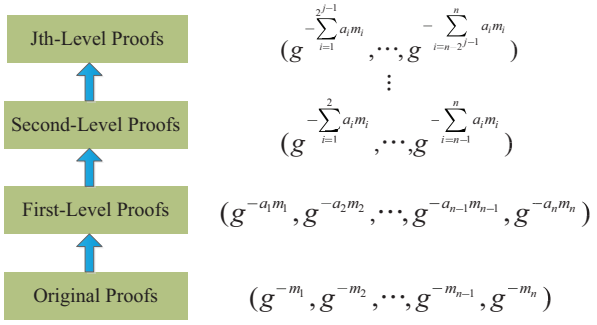


Figure 2: Aggregated proofs structure

In the PoW protocol, many scholars adopt the Merkel hash tree to verify ownership of the data. These schemes do not verify the data ownership based on accessing of the original data. In other words, the verification is based on the Merkle hash tree which is built over one hash of the original data rather than the original data. Therefore, a malicious user could pass the PoW verification of client-side deduplication if he could get the hash value of the data. However, the aggregated proofs structure verifies the data ownership by the original data block. Moreover, the proposed scheme improves the detection rate and saves network bandwidth by using the aggregated proofs structure.

4 Problem Statement

4.1 System Model

The system model of the proposed scheme is described as Figure 3, which includes five entities: cloud service provider, key server, third party auditor, first uploader and subsequent uploaders.

- **Cloud Service Provider (CSP)**: The CSP stores the encrypted data uploaded by the first uploader and performs deduplication operations with subsequent

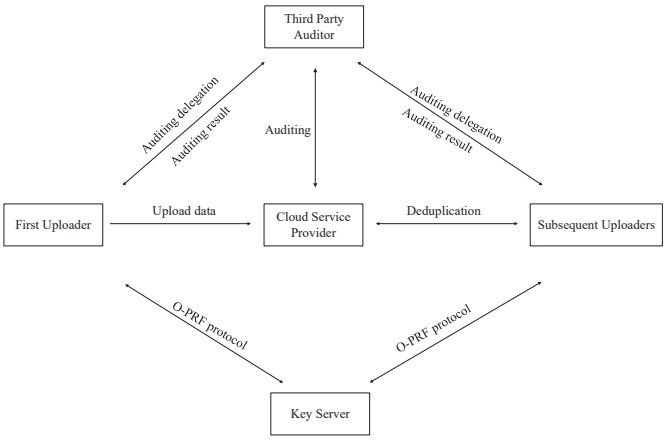


Figure 3: System model of our scheme

uploaders. In addition, when uploaders delegate a third party auditor to audit cloud storage data, the CSP generates corresponding auditing proofs.

- **Key Server (KS)**: In the O-PRF protocol, upon receiving the blinded hash value from the uploader, the KS signs it using the oblivious pseudo-random function. As a result of the interacting, the ciphertext is secure against the brute-force attack by known set.
- **Third Party Auditor (TPA)**: When the uploader sends auditing delegation, the TPA invokes the integrity auditing protocol by sending the CSP a challenge. On receipt of the proof from the CSP, TPA verifies the target data integrity and notifies the client of the result.
- **First Uploader**: The first uploader interacts with the KS to generate a secret value of the data by the O-PRF protocol. Then, the first uploader generates ciphertext and uploads it to the CSP. When the first uploader wants to check the integrity of the cloud storage data, he sends the auditing delegation to the TPA.
- **Subsequent Uploaders**: The subsequent uploaders interact with the KS to generate a secret value of the data by the O-PRF protocol. Moreover, the subsequent uploaders perform deduplication operations with CSP and send the auditing delegation to TPA.

4.2 Threat and Security Model

We give the threat model of the proof of ownership process and the key distribution process. Moreover, the security model of integrity auditing scheme is defined.

4.2.1 Threat Model

The existing deduplication schemes [6,28] did not construct a formal security model, but only gave some threat models. Therefore, the threat model for a malicious user and the cloud servers is constructed as follows.

In PoW protocol, a malicious user is to pass the PoW challenge for a message m while he only knows some partial information of m. Suppose that the malicious user possesses several data blocks and the hash value of m . Therefore, the malicious user can attempt to forge proofs for passing the PoW challenge. On the other hand, the cloud server wants to get some information about the data during the ownership verification process.

In key distribution process, a malicious user owns the secret value of the data, but he is an unauthorized user. The malicious user can attempt to get the symmetric key of the data. Moreover, since the cloud server re-encrypts the ciphertext of the data, the cloud server also attempts to obtain the symmetric key of the data.

4.2.2 Security Model

As for the security of integrity auditing, similar to the existing definition [20], we consider the probability that the cloud server can convince the user that the cloud storage data is stored correctly while the cloud storage data has been corrupted or deleted. We say that the proposed scheme is secure against an adaptive chosen-message attack. The specific process of this game is as follows.

Setup: We divide users into normal users and malicious users. For the l normal users and l' malicious users in the system, the challenger performs KeyGen algorithm to generate user-associated public/private key pairs $(pk_{nu}, sk_{nu})_{nu\in[1,l]}$ and $(pk_{mu}, sk_{mu})_{mu\in[1,l']}$. Finally, the normal users' public keys and the malicious users' public/ private keys are sent to the adversary $\mathcal{A}$.

Query 1: The adversary $\mathcal{A}$ can adaptively query **SecValGen** to obtain message-related public/private key pairs $(pk_{\pi'_\omega}, \pi'_\omega)_{\omega\in[1,o']}$ for o' data. Then, $\mathcal{A}$ queries **Rekey** and gets re-signature keys $rk'_{nu,m}$ and $rk'_{mu,m}$. Finally, $\mathcal{A}$ adaptively queries **TagBlock** as follows.

The adversary $\mathcal{A}$ chooses a block E'_1 and sends it to the challenger for the tag under message-related public key $pk_{\pi'_\omega}$. The challenger calls **TagBlock** algorithm and sends $T'_{1,\omega}$ back to $\mathcal{A}$. $\mathcal{A}$ continually queries the tags on blocks $E'_2, \ldots, E'_{n'}$ under $pk_{\pi'_\omega}$, and the challenger responds $T'_{2,\omega}, \ldots, T'_{n',\omega}$ accordingly. In the end, $\mathcal{A}$ stores the blocks and their tags.

Query 2: The adversary $\mathcal{A}$ can adaptively query **SecValGen** to obtain message-related public key $(pk_{\pi'_\omega})_{\omega\in[1,o]}$ for o data. $\mathcal{A}$ then queries **Rekey** and gets re-signature keys $rk'_{nu,m}$ for $(1 \le nu \le l, 1 \le \omega \le o)$. Finally, $\mathcal{A}$ adaptively queries **TagBlock** on blocks $E_1, E_2, \ldots, E_n$ as the case in query 1.

Challenge: The challenger requests $\mathcal{A}$ to provide a proof of possession for $\{E_i\}_{i\in I\subseteq[1,n]}$ determined by a challenge $Chal$ under the user public key pk_{nu}.

Forge: The adversary $\mathcal{A}$ outputs a possession proof P.

If CheckProof returns 1, then the adversary $\mathcal{A}$ wins this game.

Definition 3. *We say that a data integrity auditing scheme is secure, if for any probabilistic polynomial time adversary $\mathcal{A}$ who does not possess all of the challenged data blocks, the probability that $\mathcal{A}$ succeeds in the above game is negligible.*

4.3 Working Graphs

As shown in Figure 4, we describe in detail the whole process of our scheme.

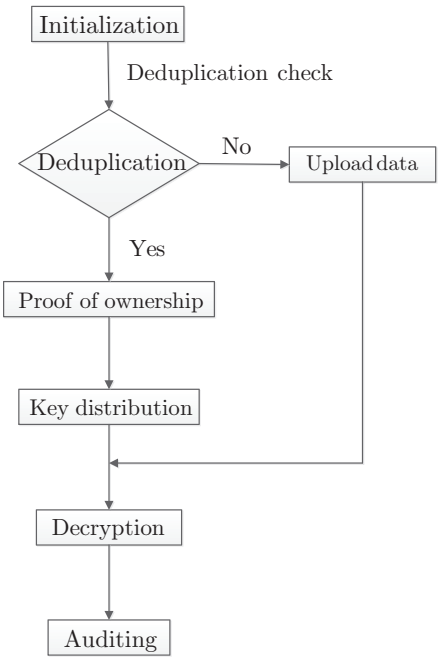


Figure 4: The working graphs of our scheme

In the initialization phase, the system generates the public parameters. The cloud server and the uploader generate their key pairs. In addition, uploaders interact with the key server to generate a secret value of the data. When a uploader uploads a tag to the cloud server, the cloud server performs deduplication detection on the data. If the data does not exist in cloud server, this uploader is the first uploader and uploads the data to the cloud server. Otherwise, this uploader is subsequent uploader and goes into the deduplication phase. The proof of ownership protocol aims to verify that the subsequent uploader indeed owns the original data. When the subsequent uploader passes the proof of ownership protocol, the cloud server distributes the symmetric key to subsequent uploader by using proxy re-encryption technology. Because the first uploader and subsequent uploader are data owners, they can decrypt the ciphertext to get the plaintext data. Moreover, the data owners authorize a third party auditor to audit the integrity of the cloud storage data.

5 Proposed Construction

In this section, we put forward a secure and efficient client-side data deduplication scheme with public auditing. The proposed scheme consists of five phases: initialization, upload, deduplication, decryption, and auditing.

5.1 The Initialization Phase

The system runs the Setup algorithm and generates the public parameters. Moreover, the CSP and the uploader generate their key pairs by running the KeyGen algorithm. In addition, uploaders interact with the key server to generate secret value of the data by running the secret value generation (SecValGen) algorithm. The details are as follows.

Setup: Let p, q be two large primes. Due to the property of safe primes, there exist two primes p' and q' that satisfy that $p = 2p' + 1$, $q = 2q' + 1$. We compute $n = p * q$ and choose generator g with order $\lambda = 2p'q'$, which can be chosen by selecting a random number $\varsigma \in Z^*_{n^2}$ and computing $g = -\varsigma^{2n}$. The value λ can be used for decryption, but we choose to conceal and protect it from all parties. In addition, the system chooses two groups G_1 and G_T of a prime order with bilinear map $e : G_1 \times G_1 \to G_T$. The system parameters are random generators $v \in G_1$ and $Z = e(v, v) \in G_T$. Then, it randomly chooses secure hash functions $H_1 : \{0,1\}^* \to Z^*_n$, $H_2 : \{0,1\}^* \to G_1$, $H_3 : Z^*_n \to \{0,1\}^*$. The system public parameters are $pars = (G_1, G_T, n, g, Z, H_1, H_2, H_3)$.

KeyGen: The CSP and the uploader j generates their key pairs: $(sk_{CSP}, pk_{CSP}) = (a, v^a)$ and $(sk_j, pk_j) = (u_j, v^{u_j})$ respectively. Besides, the uploader j selects a random number $x_{u_j} \in Z^*_n$ as his secret value.

SecValGen: The uploader interacts with the key server to generate a secret value of the data by the O-PRF protocol [21]. The O-PRF protocol is based on RSA blind signatures. The key server has secret key d and public key e where $ed \equiv 1 \bmod \Phi(n)$. $H_1 : \{0,1\}^* \to Z^*_n$ and $H_3 : Z^*_n \to \{0,1\}^*$ are two secure hash functions. The interaction process is as follows.

1) The uploader calculates the hash value $h = H_1(m)$ of the data and selects a random value $r \in Z^*_n$. Moreover, the uploader computes the blinded hash $x = h \cdot r^e \bmod n$ and sends x to the key server.

2) Upon receiving x, the key server computes $y = x^d \bmod n$ and sends y to the uploader.

3) The uploader calculates $y \cdot r^{-1} \bmod n$ and obtains secret value z. Finally, the uploader computes a secret value $\pi \leftarrow H_3(z)$.

Then, the uploader announces that it has a certain data via a tag. If the data does not exist in CSP, the uploader

goes into the upload phase. Otherwise, the uploader goes into the deduplication phase.

5.2 The Upload Phase

The first uploader performs an upload task that includes Encrypt algorithm and TagBlock algorithm. In this process, the first uploader generates the data ciphertext, the data tag, the ciphertext of symmetric key, the original proofs of data, and the auditing tag of data block. Finally, the first uploader uploads these information to the cloud server.

Encrypt: The encryption algorithm is divided into four parts, as shown below.

1) The first uploader generates the data ciphertext $E = Enc_k(m)$ using a random symmetric key k.

2) The first uploader computes the data tag $T = H_1(m)$.

3) The first uploader chooses two random values r_1 and r_2, and then encrypts symmetric key k using the public keys pk_{CSP} and the secret value π. The ciphertext of symmetric key k is denoted as: $C = \{C_1, C_2, C_3\} = \{(1 + k * n)g^{H_1(Z^{r_1})*r_2} \bmod n^2, g^{r_2} \bmod n^2, pk_{CSP}^{\pi r_1}\}$.

4) The first uploader divides the data m into n blocks (i.e., $m = \{m_1, \ldots, m_n\}$), and calculates the original proofs: $IPs_i = g^{-m_i} (\mathrm{mod} p)$, $IPs = (IPs_1, \ldots, IPs_n)$.

TagBlock: The first uploader computes data integrity tags. In particular, he splits E into n blocks (i.e., $E = \{E_1, E_2, \ldots, E_n\}$) and further divides each block E_i into s sectors (i.e., $E_i = \{E_{i,1}, E_{i,2}, \ldots, E_{i,s}\}$). Among them $g_1, g_2, \ldots, g_s$ are s elements in G_1. For each block E_i, the first uploader computes $T_i = [H_2(ID_C \parallel i) \cdot \prod_{j=1}^{s} g_j{}^{E_{i,j}}]^{\pi} \in G_1$.

5.3 The Deduplication Phase

If a data announced by the first uploader in the initialization phase exists in the cloud server, the subsequent uploaders go into the deduplication phase and run the proof of ownership protocol. When the subsequent uploader passes the proof of ownership protocol, the cloud server generates the re-encryption key and re-encrypts the ciphertext of symmetric key.

5.3.1 The Proof of Ownership Protocol

The proof of ownership protocol aims to provide a framework for the cloud server to verify that the subsequent uploader indeed owns the data rather than part of it. This phase includes the original proofs generation (OPsGen) algorithm, the coefficients generation (CosGen) algorithm, the final proofs generation (FPsGen) algorithm, and the proofs verification (ProVer) algorithm.

OPsGen: As shown in the encryption process, the first uploader generates the original proofs of the data: $IPs_i = g^{-m_i} (\mathrm{mod} p)$, $IPs = (IPs_1, \ldots, IPs_n)$. Then he sends these proofs to the cloud server. After receiving the original proofs, the cloud server aggregates these proofs using the aggregated proofs structure. As shown in Subsection 3.4.

CosGen: The subsequent uploaders utilize this algorithm to generate the coefficients according to the Schnorr's Identification Protocol [28]. Given μ random number r_i, $1 \le r_i \le q-1$, the subsequent uploaders compute the Coefficients: $commit_i = g^{r_i} (\mathrm{mod} p)$, $commit = (commit_1, \ldots, commit_\mu)$, and send these Coefficients to the cloud server.

FPsGen: The subsequent uploaders run this algorithm to generate the final proofs for cloud server's challenge. Let $Chal = (\theta, \gamma, a_1, a_2, \ldots, a_n)$, $1 \le \gamma \le 2^\alpha$ be selected uniformly at random, μ be the commit generated by Coefficients and $\theta, a_1, a_2, \ldots, a_n$ are a set of random integers. The cloud server allows the subsequent uploader to challenge the jth-level aggregated proofs and sends the challenge block index function $\varphi_\theta(\cdot)$ to the subsequent uploader. According to the block index function and index number, the subsequent uploader calculates the challenge set of the data. Then, the subsequent uploader calculates the final proofs:

$$\begin{aligned} FPs_t &= \gamma \sum_{i=1}^{2^{j-1}} a_i m_i + r_t (\mathrm{mod} q), \\ FPs &= (FPs_1, \ldots, FPs_{\lfloor \frac{n}{2^{j-1}} \rfloor}). \end{aligned}$$

ProVer: The cloud server receives the final proofs FPs from the subsequent uploader. According to the aggregated proofs structure, challenge set $Chal$ and Coefficients $commit$, the cloud server computes the product representation of the IPs and FPs: $DPs_t = g^{FPs_t} \times IPs_t{}^{\gamma} (\mathrm{mod} p)$. If $DPs_t = commit_t$, output true, the proof is recognized by the cloud server. Otherwise, output false, the proof is fake.

5.3.2 The Key Distribution Process

The cloud server generates the re-encryption key and the re-encryption ciphertext. This process includes the re-encryption key generation (RekGen) algorithm and the re-encryption (ReEnc) algorithm.

RekGen: The cloud server wants to delegate the subsequent uploader j by publishing re-encryption key $rk_{CSP \to j} = v^{u_j / a}$.

ReEnc: The cloud server computes ciphertext $C_3' = e(pk_{CSP}^{\pi r_1}, rk_{CSP \to j}) = Z^{\pi r_1 * u_j}$, and sets $C_2' = C_2$ and $C_1' = C_1$. Finally, the cloud server generates the re-encryption ciphertext $C' = \{C_1', C_2', C_3'\}$.

5.4 The Decryption Phase

When the data owner proposes a decryption request, the cloud server sends the ciphertext to the data owner. Then, the Decrypt algorithm is as follows.

Decrypt: Upon receiving the encrypted data tuple (E, C'), the data owner can directly decrypt it to obtain the original data. The specific steps for decryption are as follows.

1) The data owner computes $C_3'' = H_1((C_3')^{1/\pi u_j}) = H_1(Z^{r_1})$.

2) The data owner obtains the symmetric key $k = L(C_1/(C_2')^{C_3''} \bmod n^2)$ where $L(u) = (u-1)/n$.

3) The data owner obtains the data $m = Dec_k(E)$ using the symmetric key k.

5.5 The Auditing Phase

During the auditing process, the proposed scheme adopts the proxy re-signature technology to achieve efficient auditing. Firstly, the user computes re-signature keys. Secondly, the cloud service provider generates corresponding auditing proofs by using cloud storage data and re-signature keys. Finally, the third party auditor verifies the integrity of the target data by the user's public key and auditing proofs. This process includes the re-signature keys generation (Rekey) algorithm, the generation proof (GenProof) algorithm, and the check proof (CheckProof) algorithm.

Rekey: It is performed by user to compute re-signature keys, which enables the cloud to prove the integrity of the challenged data under user-associated private/public key pair. The user j computers $d_{u,m} = u_j \cdot (\pi)^{-1} + \beta$, $h_{u,m} = \beta \cdot x_{u_j}$ and also sets $rk_{u,m} = (d_{u,m}, h_{u,m})$, where β is a random number in Z_n^*.

GenProof: The third party auditor chooses a random c-element subset $I \subset [1, n]$ along with c random coefficients in Z_n^*. Let $Q = \{i, v_i\}_{i \in I}$ be the set of challenge index-coefficient pairs. After receiving Q from the third party auditor, the cloud server sends a proof $P = (\sigma, \sigma_1, \rho_1, \ldots, \rho_s)$ back to the third party auditor, where $\sigma = \prod_{(i,v_i)\in Q} T_i^{d_{u,m}\cdot v_i} \in G_1$, $\sigma_1 = \prod_{(i,v_i)\in Q} T_i^{h_{u,m}\cdot v_i} \in G_1$ and $\rho_j = \sum_{(i,v_i)\in Q} v_i \cdot E_{i,j} \in Z_n^*$ for $1 \le j \le s$.

CheckProof: The third party auditor sends σ_1 to user and obtains $\sigma_1' = \sigma_1^{x_u^{-1}}$. The third party auditor then accepts the proof if the following equation holds:

$$e(\frac{\sigma}{\sigma_1'}, g) = e(\prod_{(i,v_i)\in Q} H_2(ID_C \parallel i)^{v_i} \cdot \prod_{j=1}^{s} g_j^{\rho_j}, pk_u)$$

6 Security Analysis

In this section, we analyze security of the proposed scheme. The security consists of two parts: The data deduplication phase and the data auditing phase.

6.1 Data Deduplication Phase

In this case, we mainly concentrate on the security of the PoW protocol, the O-PRF protocol, and the ciphertext of symmetric key.

Theorem 1. *The proposed proofs of ownership (PoW) protocol is a zero-knowledge proof of knowledge assuming that the discrete logarithm is hard.*

Proof. A zero-knowledge proof protocol satisfies the following three properties: completeness, soundness and zero-knowledge. Assuming that the discrete logarithm problem is hard means that no adversary can compute the secret value m_i from the original proofs IPs_i, where $IPs_i = g^{-m_i} (\mathrm{mod} p)$. In the process of aggregating proofs, the cloud server aggregates the original proofs into the j-level proofs. For the j-level proofs, no adversary can compute the secret information m_i. □

Completeness. Completeness means that a client has the original data blocks, and both the client and cloud server follow the instructions, then the cloud server must accept the client. This is because

$$\begin{aligned} DPs_t &= g^{FPs_t} \times (IPs_t)^{\gamma} \bmod p \\ &= g^{\gamma \sum_{i=1}^{2^{j-1}} a_i m_i + r_t} \cdot (g^{-\sum_{i=1}^{2^{j-1}} a_i m_i})^{\gamma} \bmod p \\ &= g^{r_t} \bmod p \\ &= commit_t \end{aligned}$$

Soundness. Soundness means that if a client does not have the original data blocks, then regardless of what the client does, the cloud server will pass the proofs with probability that it can be ignored. Assuming the client is a cheater, he does not have the correct original data blocks m_i. $commit_t$ is transmitted in iteration, the server, after picking $\gamma \in \{0,1\}^{\alpha}$, is waiting for:

$$FPs_t = \log_g(commit_t IPs_t{}^{\gamma} \bmod p) (\mathrm{mod} q)$$

This equation shows that, for fixed $commit_t$ and IPs_t, there will be 2^{α} distinct values for FPs_t which correspond to 2^{α} distinct values for e. So the client guesses probability for each $-\sum_{i=1}^{2^{j-1}} a_i m_i$ is $2^{-\alpha}$. Here let $\lfloor \frac{n}{2^{j-1}} \rfloor$ be equal to η. In PoW protocol, the client interacts with the server η times. If all the η commitments are admitted, the cloud server marks this client as the owner of this data. The false positive probability for the verify protocol is $2^{-\eta\alpha}$.

Zero-knowledge. For a perfect zero-knowledge proof protocol, which does not need to negotiate between

the prover and the verifier. We introduce a simulator that produces the proof transcript of simulation. During the proof of ownership of this document, the simulator effectively generates the proof transcript without interacting with the real client, and the transcript generated by these simulators is indistinguishable from the actual transcript.

For common input IPs_t, we can construct a polynomial-time (in $|p|$) simulator S as follows.

1) S initializes transcript as an empty string;
2) (a) S picks $FPs_t \in Z_q$; (b) S picks $\gamma \in \{0,1\}^\alpha$; FPs_t must be uniform in Z_q for either cases of $\gamma \in \{0,1\}^\alpha$ and independent of the common input IPs_t; (c) S computes $commit_t \leftarrow g^{FPs_t} IPs_t{}^\gamma \bmod p$; $commit$ must also be uniform and independent of the common input IPs_t; (d) $Transcript \leftarrow Transcript \parallel commit_t, \gamma, FPs_t$.

Clearly, $Transcript(commit_t, \gamma, FPs_t)$ can be produced by S in polynomial time, and the elements in it have distributions which are the same as those in a real proof transcript. Therefore, the protocol is perfect zero-knowledge.

In summary, the data sent from the client in a run is uniform, they can tell the cloud server that there is no information about the client's private input b_i. Regardless of how the server selects the random challenge bits, the elements in the client's records are uniform, so even if the cloud server is dishonest, the protocol is a perfect zero knowledge.

Theorem 2. *The O-PRF protocol is an interactive protocol between the uploader and the key server. The key server will not obtain the secret value π. In addition, during the O-PRF protocol, no information will be revealed.*

Proof. In the O-PRF protocol, the uploader blinds the hash value $H_1(m)$ of the data and sends it to the key server which can not obtain the hash value $H_1(m)$ of data. Therefore, the key server will not obtain the secret value π. In the process of interaction between the uploader and the key server, the uploader blinds the hash value $H_1(m)$ and sends x to the key server. The key server signs the blinded value x and sends y to the uploader. Because x and y are blinded by the random value r, the O-PRF protocol will not leak any information. □

Theorem 3. *If the DL problem holds in group G_1 and the CDH problem holds in group $Z^*_{n^2}$, then the ciphertext of symmetric key is secure in the proposed scheme.*

Proof. The ciphertext of the symmetric key is $C = \{C_1, C_2, C_3\} = \{(1 + k * n) g^{H_1(Z^{r_1}) * r_2} \bmod n^2, g^{r_2} \bmod n^2, pk_{CSP}^{\pi r_1}\}$. The cloud server and unauthorized users would like to obtain the symmetric key k. □

The secret value π is obtained by the O-PRF protocol between the uploader and the key server. According to Theorem 2, the cloud server can not obtain the secret value π. Because the DL problem is difficult, it is hard to get v^{r_1} from $pk_{CSP}^{\pi r_1} = v^{\pi r_1 a}$. Thus, the cloud server can not obtain the value of $H(Z^{r_1})$. The cloud server re-encrypts the ciphertext of the symmetric key, the obtained ciphertext is: $C' = \{C'_1, C'_2, C'_3\} = \{(1 + k * n) g^{H_1(Z^{r_1}) * r_2} \bmod n^2, g^{r_2} \bmod n^2, Z^{\pi r_1 * u_j}\}$. Unauthorized users with a secret value of π also can not obtain the value $H(Z^{r_1})$, because he can not obtain the private key u_j of user j. Bounded by the difficulty of the CDH problem, the cloud server and unauthorized users can not get $g^{H_1(Z^{r_1}) * r_2}$ from $g^{H_1(Z^{r_1})}$ and g^{r_2}. Hence, they can not obtain the symmetric key k. In addition, a malicious user who does not have a secret key d, therefore, cannot generate secret value π. It allows encryption to be secure against the brute force attacks even for predictable message set. Therefore, the key distribution of the proposed scheme is secure.

6.2 Data Auditing Phase

In this case, we focus on the correctness and unforgeability of the integrity auditing scheme.

Theorem 4. *The cloud server is able to generate a proof that passes the verification if all the challenged blocks and their integrity tags are correctly stored.*

Proof. Proving the correctness of our integrity auditing scheme for data is equivalent to proving that equation $e(\frac{\sigma}{\sigma'_1}, g) = e(\prod_{(i,v_i)\in Q} H_2(ID_C \parallel i)^{v_i} \cdot \prod_{j=1}^{s} g_j^{\rho_j}, pk_u)$ hold. According to the properties of the bilinear map, the correctness can be verified by the following calculations.

$$
\begin{aligned}
& e(\tfrac{\sigma}{\sigma'_1}, g) \\
&= e(\textstyle\prod_{(i,v_i)\in Q} T_i^{d_{u,m} v_i} \cdot (\prod_{(i,v_i)\in Q} T_i^{r_{u,m} v_i})^{-1}, g) \\
&= e(\textstyle\prod_{(i,v_i)\in Q} T_i^{u_j \pi^{-1} v_i}, g) \\
&= e(\textstyle\prod_{(i,v_i)\in Q} [H_2(ID_C \parallel i) \cdot \prod_{j=1}^{s} g_j^{E_{i,j}}]^{u_j \cdot v_i}, g) \\
&= e(\textstyle\prod_{(i,v_i)\in Q} [H_2(ID_C \parallel i)^{v_i} \cdot \prod_{j=1}^{s} g_j^{v_i E_{i,j}}], pk_u) \\
&= e(\textstyle\prod_{(i,v_i)\in Q} [H_2(ID_C \parallel i)^{v_i} \cdot \prod_{j=1}^{s} g_j^{\rho_j}], pk_u)
\end{aligned}
$$

□

Theorem 5. *Under the CDH assumption, the integrity auditing scheme is secure against an adaptive chosen-message attack in the random oracle model.*

Proof. Assuming that the CDH assumption holds in G. If there is a polynomial time adversary $\mathcal{A}$, he has the advantage $Adv_{\mathcal{A}}$ to break our scheme. Then, we show how to construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to solve the CDH problem. That is, given a CDH tuple (g, g^a, g_0), the adversary $\mathcal{B}$ is able to compute g_0^a with non-negligible probability. In the process of proof, the adversary $\mathcal{B}$ is the challenger for the adversary $\mathcal{A}$. The process of proof is as follows. □

Setup: The normal user-associated public keys are set to be $pk_{nu} = g^{a s_{nu}}$ for $nu \in [1, l]$, where s_{nu} are randomly chosen from Z_q^*. Moreover, the adversary

$\mathcal{B}$ sets $g_j = g_0^{y_j}$ for $1 \le j \le s$. Besides, $\mathcal{B}$ chooses x_{nu} randomly from Z_q^*. For malicious users, $\mathcal{B}$ selects random numbers x_{mu}, s_{mu} $(mu \in [1, l'])$ and computes the public keys $pk_{mu} = g^{s_{mu}}$. Finally, the system parameters, the normal user public keys $pk_{nu}(nu \in [1, l])$, the malicious public and private key pairs $(pk_{mu}, s_{mu}, x_{mu})(mu \in [1, l'])$ are given to the adversary $\mathcal{A}$.

Query 1: There are four types of queries that $\mathcal{A}$ can request: oracle SecValGen, oracle Rekey, oracle TagBlock and the hash function H_1.

1) Oracle **SecValGen:** if π'_ω has not been queried before, $\mathcal{B}$ returns a random number $x'_w \in Z_q^*$ to $\mathcal{A}$ and records it in list DataKey. Otherwise, $\mathcal{B}$ obtains x'_w from list DataKey and responds it to $\mathcal{A}$.

2) Oracle **Rekey:** for malicious user public key pk_{mu}, the adversary $\mathcal{B}$ returns $(x'_w)^{-1} \cdot s_{mu} + r_{mu,w'} \cdot x_{mu}$, where $r_{mu,w'}$ is a random in Z_q^*. For normal user, $\mathcal{B}$ returns two random numbers to $\mathcal{A}$.

3) Oracle **TagBlock:** if $ID_{E'_i} \parallel V_i$ has not been queried before, the adversary $\mathcal{B}$ chooses a random element from G_1 as the value of $H_1(ID_{E'_i} \parallel V_i)$ and then computes $T'_i = [H_1(ID_{E'_i} \parallel V_i) \cdot \prod_{j=1}^{s} g_j^{E_{i,j}}]^{x'_w}$ for the query. Finally, $\mathcal{B}$ records T'_i in list and returns $H_1(ID_{E'_i} \parallel V_i)$ for the corresponding hash query. Otherwise, the adversary $\mathcal{B}$ returns T'_i from list to the adversary $\mathcal{A}$.

Query 2: There are three types of queries that $\mathcal{A}$ can request: oracle Rekey, oracle TagBlock and the hash function H_1.

1) Oracle **Rekey:** The adversary $\mathcal{B}$ returns $(x_w^{-1} \cdot s_{nu} + r_{nu,w}, x_{nu} r_{nu,w})$ to $\mathcal{A}$, where $r_{nu,w}$ is a random number in Z_q^*.

2) Oracle **TagBlock:** if $ID_{E'_i} \parallel V_i$ has not been queried before, the adversary $\mathcal{B}$ computes $T_i = g^{a x_w r_i}$ and records it in list. Finally, $\mathcal{B}$ returns T_i and $H_1(ID_{E'_i} \parallel V_i) = \frac{g^{r_i}}{\prod_{j=1}^{s} g_j^{E_{i,j}}}$ for the corresponding hash query. It is easily observed that T_i is a valid tag under the public key pk_π. If $\{E_i, ID_{E_i} \parallel V_i\}$ is in list, the adversary $\mathcal{B}$ obtains T_i and returns it to the adversary $\mathcal{A}$.

Challenge: The adversary $\mathcal{B}$ requests the adversary $\mathcal{A}$ to prove the integrity of all blocks $E_1, \ldots, E_n$ by sending coefficients $a_1, \ldots, a_n$ under the public key pk_u.

Forge: We assume that the adversary $\mathcal{A}$ has deleted or modified one or more blocks. Let $\rho'_j = \sum_{i=1}^{n} a_i E_{i,j}$ be the real result. The adversary $\mathcal{A}$ returns a proof $P = (\sigma, \sigma_1, \rho_1, \ldots, \rho_s)$ satisfying $e(\frac{\sigma}{\sigma'_1}, g) = e(\prod_{(i,v_i) \in Q} H_2(ID_C \parallel i)^{v_i} \cdot \prod_{j=1}^{s} g_j^{\rho_j}, pk_u)$ but there exists at least one value $\rho_j = \rho'_j$. Since P is a valid proof under public key pk_u, we have

$$\begin{aligned}\frac{\sigma}{\sigma'_1} &= [\prod_{i=1}^{n} H_1(ID_C \parallel V_i)^{a_i} \cdot \prod_{j=1}^{s} g_j^{\rho_j}]^{a s_u} \\ &= (\prod_{i=1}^{n} (H_1(\frac{g^{r_i}}{\prod_{j=1}^{s} g_j^{m_{i,j}}})^{a_i} \cdot \prod_{j=1}^{s} g_j^{\rho_j})^{a s_u} \\ &= (\prod_{i=1}^{n} g^{r_i a_i} \cdot \prod_{j=1}^{s} g_j^{\rho_j - \rho'_j})^{a s_u} \\ &= g^{a s_u \sum_{i=1}^{n} r_i a_i} (g_0^{s_u \sum_{j=1}^{s} y_j (\rho_j - \rho'_j)})^a\end{aligned}$$

From the above equation, the adversary $\mathcal{B}$ can easily compute $g_0^a = (\frac{\sigma}{\sigma'_1 g^{a s_u \sum_{i=1}^{n} r_i a_i}})^{[s_u \sum_{j=1}^{s} y_j (\rho_j - \rho'_j)]^{-1}}$.

If the adversary $\mathcal{A}$ does not possess all the sectors $E_{i,j}$ $(1 \le i \le n, 1 \le j \le s)$, we analyze the probability that the adversary $\mathcal{A}$ successfully forges the values satisfying $P_j = P'_j$ for $(1 \le j \le s)$. Due to Theorem 2 in [20], we can know that the adversary $\mathcal{A}$ forges a valid value $P_j = P'_j$ is negligible.

Finally, if there is a polynomial time adversary $\mathcal{A}$ that has the advantage $Adv_{\mathcal{A}}$ to break our scheme, the adversary $\mathcal{B}$ can use $\mathcal{A}$ to solve the CDH problem. Since the CDH problem is a difficult problem, the probability that the adversary $\mathcal{A}$ breaks our scheme is negligible. Therefore, the proposed scheme is secure against an adaptive chosen-message attack in the random oracle model under the CDH assumption.

7 Performance Evaluation

In this section, we will conduct the performance evaluation including four aspects, the detection rate analysis, functionality comparison, efficiency comparison, and experimental comparison.

7.1 Detection Rate Analysis

Since Ding *et al.*'s scheme [6] and Liu *et al.*'s scheme [20] do not verify the data ownership based on the original data, we only make the detection rate analysis between Yang *et al.*'s scheme [28] and our scheme.

Suppose a client claims the ownership of an n-block data m, but actually he owns f out of n blocks of data m. Let's examine the probability that the cloud server accepts the client as the data owner. We use p_x to indicate the probability that the cloud server detects at least one missing block. We set x as the number of missing data blocks. During the proof of ownership process, if the missing data block on the client is not detected, the cloud server will accept the client's ownership of the data. Therefore, the probability that the client is accepted by the cloud server is $1 - p_x$. Since

$$p_x = p\{x \ge 1\} = 1 - \frac{C_{n-x}^{\mu}}{C_n^{\mu}} = 1 - \prod_{i=0}^{\mu-1} \frac{n-x-i}{n-i}$$

Based on the knowledge of probability theory, we can calculate: $1 - p_x \approx (1 - \frac{x}{n})^{\mu}$.

From the above equation, we can derive: $\mu \approx \left\lceil \log_{(1-\frac{x}{n})}(1 - p_x) \right\rceil$.

Table 1: Functionality comparison between our scheme and other schemes

Schemes	Ding *et al.* [6]	Liu *et al.* [20]	Yang *et al.* [28]	Our scheme
Secure proof of ownership	No	No	Yes	Yes
High detection rate	No	No	No	Yes
Data owner offline	Yes	No	No	Yes
No authorized party	No	No	Yes	Yes
Against brute-force attacks	Yes	No	Yes	Yes
Public auditing	No	Yes	No	Yes
One tag for each block	No	Yes	No	Yes

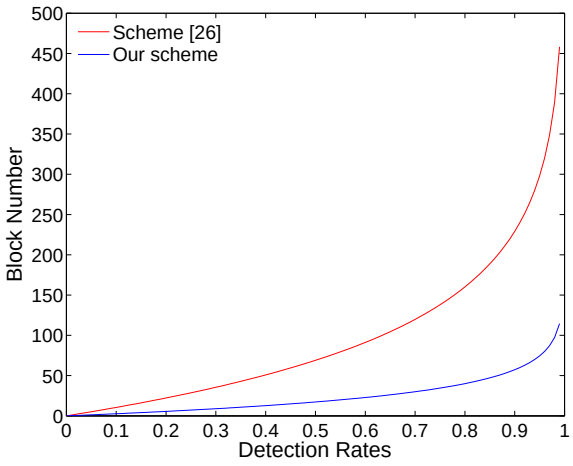


Figure 5: Challenged block numbers vary with detection rates

The proposed scheme aggregates original proofs and verifies the subsequent proofs. With the same number of verifications, we compare the detection rate of our scheme and Yang *et al.*'s scheme [28]. Figure 5 is the relationship between the detection rate and the number of challenges in our scheme and Yang *et al.*'s scheme [28] in the case of missing blocks rates of 1%. In the proposed scheme, we utilize the third-level aggregate proofs to verify ownership of the data. It can be seen that the detection rate of the proposed scheme is higher than Yang *et al.*'s scheme [28] with the same of data blocks.

7.2 Functionality Comparison

The functionality comparisons between the proposed scheme and the related schemes [6, 20, 28] are showed in Table 1.

Table 1 shows that the proposed scheme supports secure proof of ownership, high detection rate, data owner offline, no authorized party, against brute-force attacks and public auditing, while others only support partial functionality. By adopting the zero-knowledge proof technology and the aggregated proofs structure, the proposed scheme achieves the secure proof of ownership and high detection rate for the missing data block, respectively. Meanwhile, we combine the oblivious pseudo-random function protocol with proxy re-encryption technology to implement key distribution without online data owners or the authorized party. In addition, because the O-PRF protocol is used in the encryption process, the proposed scheme is secure against brute-force attacks. The proposed scheme utilizes a third party auditor for performing public auditing. Moreover, our scheme only generates one tag for each data block.

7.3 Efficiency Comparison

In this subsection, we will conduct efficiency comparisons including three aspects, the proof of ownership, the key distribution, and the public auditing. Since Liu *et al.*'s scheme [20] does not support client-side deduplication, we only make a comparison between the proposed scheme and the related schemes [6, 28] in proof of ownership and key distribution processes, respectively. Moreover, we conduct a comparison between the proposed scheme and Liu *et al.*'s scheme [20] in public auditing process, because Yang *et al.*'s scheme [28] and Ding *et al.*'s scheme [6] do not support public auditing.

As shown in the table below. $Pair$: bilinear pairing; Exp: exponentiation in G_1 or G_T; $ModExp$: modular exponentiation; $ModMul$: modular multiplication; n: the number of blocks; μ: the number of challenging blocks by the cloud server; j: the level of aggregated proofs; c: the number of subsequent uploaders; s: the number of sectors for each block; d: the number of challenging blocks by the third party auditor.

7.3.1 The Proof of Ownership Process

We make an efficiency comparison between the proposed scheme and the related schemes [6, 28] in Table 2. In ownership verification process, the proposed scheme and Yang *et al.*'s scheme [28] use zero knowledge proof technology, and Ding *et al.*'s scheme [6] utilizes bilinear pairing operation.

As shown in Table 2, the computation consumption of Ding *et al.*'s scheme [6] is much smaller than Yang *et al.*'s scheme [28] and the proposed scheme. However, Ding *et al.*'s scheme [6] verifies the data ownership based on the hash value. In other words, a malicious user could pass the PoW verification of client side deduplication if he could

Table 2: Efficiency comparison in proof of ownership

Schemes	Ding *et al.* [6]	Yang *et al.* [28]	Our scheme
OPs computation	Exp	$nModExp$	$nModExp$
FPs computation	Exp	$\mu ModMul$	$\frac{\mu}{j}ModMul$
Verification of μ blocks	$Pair$	$2\mu Exp+\mu ModMul$	$\frac{2\mu}{j}Exp+\frac{\mu}{j}ModMul$
Total computational costs	$2Exp+Pair$	$2\mu Exp+nModExp+2\mu ModMul$	$\frac{2\mu}{j}Exp+nModExp+\frac{2\mu}{j}ModMul$

get the hash value of the data. In addition, because of adopting the aggregated proofs structure, the computation consumption of the proposed scheme is smaller than Yang *et al.*'s scheme [28].

7.3.2 The Key Distribution Process

We make an efficiency comparison between the proposed scheme and the related schemes [6, 28] in Table 3 with regard to first uploader, CSP, subsequent uploader and AP. For the three comparison schemes, they use a symmetric encryption algorithm to encrypt the data, and then distribute the symmetric keys through the O-PRF protocol and proxy re-encryption technology. When comparing the efficiency of the three schemes, we ignore the symmetric encryption.

As shown in Table 3, Yang *et al.*'s scheme [28] incurs higher computation overhead than the proposed scheme and Ding *et al.*'s scheme [6] in the total computational costs. Meanwhile, the first uploader has a large computational overhead in Yang *et al.*'s scheme [28]. In addition, we also compare our scheme with Ding *et al.*'s scheme [6]. Although the computational cost of our scheme is slightly higher than Ding *et al.*'s scheme [6], our scheme does not require the introduction of an authorized party to complete key distribution. In the proposed scheme, the cloud server takes on the main computational costs and other entities have a little computational costs. As we all know, the cloud server's computing power can be considered infinitely, so the proposed scheme is more practical and effective.

7.3.3 The Public Auditing Process

Because Ding *et al.*'s scheme [6] and Yang *et al.*'s scheme [28] do not support public auditing, we only make an efficiency comparison between the proposed scheme and Liu *et al.*'s scheme [20]. In public auditing process, the proposed scheme and Liu *et al.*'s scheme [20] adopt the proxy re-signature technology to verify integrity of the cloud storage data. Moreover, these two schemes only generate one auditing tag for each data block.

As shown in Table 4, the computation consumption of the proposed scheme is higher than Liu *et al.*'s scheme [20]. However, because the proposed scheme utilizes the O-PRF protocol to generate secret values for calculating auditing tag, the proposed scheme can achieve better security.

7.4 Experimental Comparison

By utilizing the Pairing Based Cryptography (PBC) Library, an efficiency experiment result is given under the Linux environment. The following experiments run on a personal computer with its configuration parameters as Intel Core i5 2.5 GHz Processor and 4 GB RAM. The number of subsequent uploaders range from 10 to 50. The experiment includes five aspects, the computation cost of the first uploader, the CSP, the subsequent uploader, the AP, and the total computation cost. The experiment result given below comes from the average of 50 experiments.

As shown in Figure 6, we first evaluate the computation cost of the first uploader in key distribution process. With the same number of subsequent uploaders, the time cost of Ding *et al.*'s scheme [6] and our scheme is much less than Yang *et al.*'s scheme [28]. Figure 7 indicates that Ding *et al.*'s scheme [6], Yang *et al.*'s scheme [28] and our scheme have almost the same time overhead. In addition, we can see that the time cost of Yang *et al.*'s scheme [28] is much more than Ding *et al.*'s scheme [6] and our scheme in Figure 8. Since Yang *et al.*'s scheme [28] and our scheme do not introduce an authorized party to complete key distribution, we only show the time cost of Ding *et al.*'s scheme [6] in Figure 9. As shown in Figure 10, we compare the total computation cost in key distribution process. The computation time of Yang *et al.*'s scheme [28] is far more than Ding *et al.*'s scheme [6] and our scheme. Moreover, during the key distribution process, Ding *et al.*'s scheme [6] and Yang *et al.*'s scheme [28] require the assistance of online data owners and the authorized party, respectively. Therefore, our scheme is secure and efficient in the key distribution process.

8 Conclusions

In this paper, we have proposed a secure and efficient client side deduplication scheme with public auditing. We utilize zero-knowledge proof and aggregates proofs structure to achieve high detection rate of client missing blocks. Meanwhile, the proposed scheme achieves key distribution by the O-PRF protocol and proxy re-encryption technology. In addition, all data owners of the proposed scheme can audit cloud storage data by employing proxy re-signing technology. The security analysis shows that the proof of ownership scheme is sound, complete and zero-knowledge.

Table 3: Efficiency comparison in key distribution

Entities	Algorithm	Ding *et al.* [6]	Yang *et al.* [28]	Our scheme
First uploader	Setup	$1Exp$	$2Exp$+$1Pair$	$1Exp$
	Data upload	$2Exp$+$2ModExp$	$2cExp$	$2Exp$+$4ModExp$
	Rekey generation	–	$cExp$	–
CSP	System setup	–	–	$1Exp$
	Re-encryption	$cPair$	$cPair$	$cPair$
	Rekey generation	–	–	$cExp$
Subsequent uploaders	System setup	$cExp$	$2cExp$+$cPair$	$cExp$
	Decrypt ciphertext	$cExp$+$cModExp$	$cExp$	$cExp$+$3cModExp$
AP	System setup	$1Exp$	–	–
	Rekey generation	$cExp$	–	–
Total computational costs		$(c+2)ModExp$+ $(3c+4)Exp$+$cPair$	$(6c+2)Exp$+ $(2c+1)Pair$	$(3c+4)ModExp$+ $(3c+4)Exp$+$cPair$

Table 4: Efficiency comparison in public auditing

Schemes	Liu *et al.* [20]	Our scheme
Tag computation	$n(s+1)Exp$	$n(s+1)Exp$+$2ModExp$
Proof computation	$2dExp$	$2dExp$
Check proof	$d(s+1)Exp$+$2Pair$	$d(s+1)Exp$+$2Pair$
Total computational costs	$(ns+n+ds+3d)Exp$+$2Pair$	$(ns+n+ds+3d)Exp$+$2Pair$+$2ModExp$

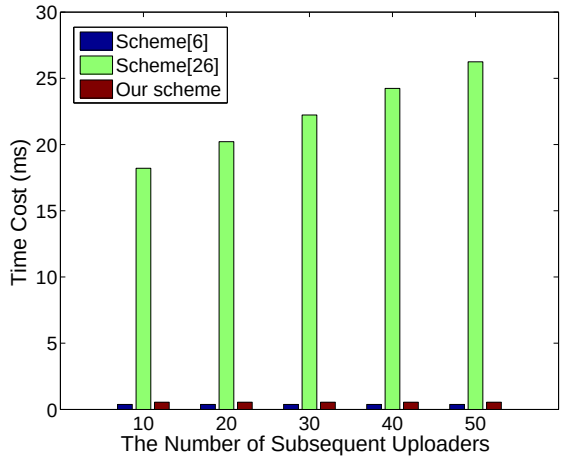


Figure 6: The computation cost of the first uploader

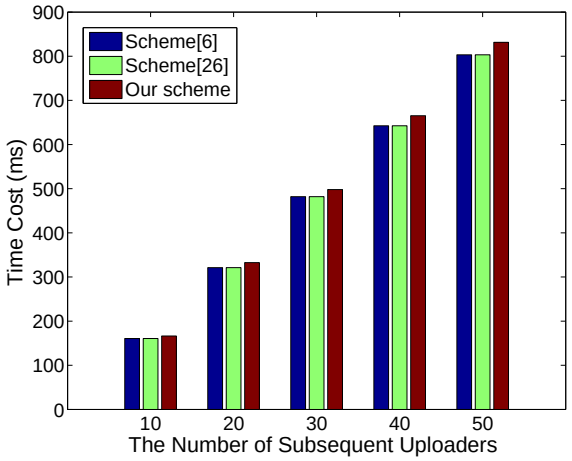


Figure 7: The computation cost of the CSP

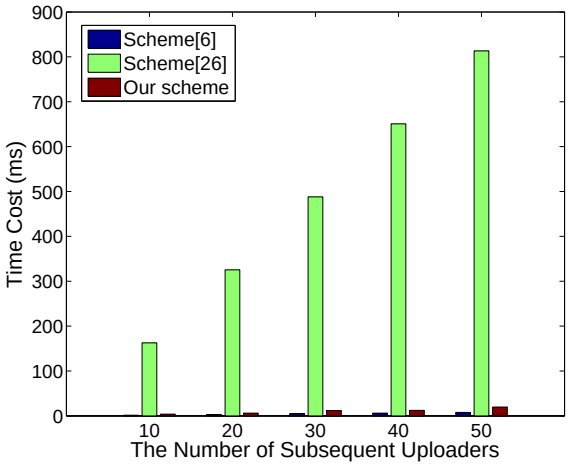


Figure 8: The computation cost of the subsequent uploaders

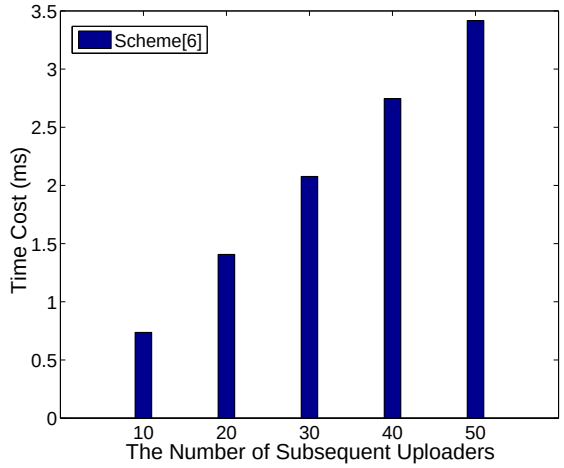


Figure 9: The computation cost of the AP

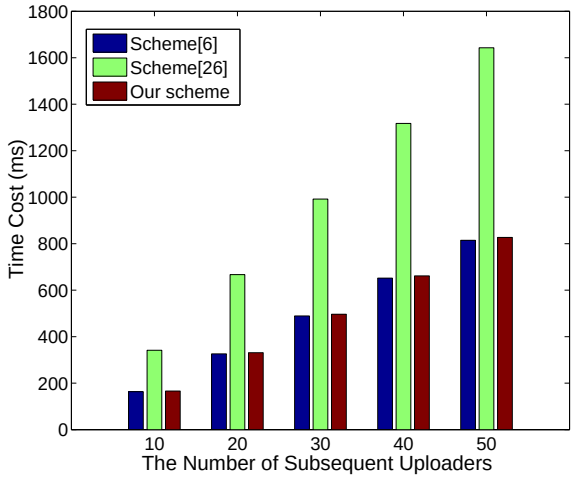


Figure 10: The total computation cost

Our scheme can protect the clients' symmetric key from being recovered by the server and other collusive clients for key distribution. In addition, our auditing scheme demonstrates the correctness and unforgeability. Finally, performance evaluation shows that the proposed scheme is practical and efficient.

Acknowledgments

We are grateful to the anonymous reviewers for their invaluable suggestions. This work is supported by the National Natural Science Foundation of China under Grants No.61472470 and 61702401.

References

[1] A. Agarwala, P. Singh, and P. K. Atrey, "Dice: A dual integrity convergent encryption protocol for client side secure data deduplication," in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 2176–2181, Oct. 2017.

[2] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: server-aided encryption for deduplicated storage," in *Usenix Conference on Security*, pp. 179–194, Aug. 2013.

[3] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 7881, pp. 296–312, 2013.

[4] Z. Cao, L. Liu, and O. Markowitch, "Analysis of one scheme for enabling cloud storage auditing with verifiable outsourcing of key updates," *International Journal of Network Security*, vol. 19, no. 6, pp. 950–954, 2017.

[5] S. Deshpande and R. Ingle, "Evidence based trust estimation model for cloud computing services," *International Journal of Network Security*, vol. 20, no. 2, pp. 291–303, 2018.

[6] W. Ding, Z. Yan, and R. H. Deng, "Secure encrypted data deduplication with ownership proof and user revocation," in *International Conference on Algorithms and Architectures for Parallel Processing*, pp. 297–312, Aug. 2017.

[7] C. C. Erway, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information and System Security (TISSEC'15)*, vol. 17, no. 4, pp. 1–29, 2015.

[8] L. Gonzalez-Manzano and A. Orfila, "An efficient confidentiality-preserving proof of ownership for deduplication," *Journal of Network and Computer Applications*, vol. 50, pp. 49–59, 2015.

[9] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *ACM Conference on Computer and Communications Security*, pp. 491–500, Oct. 2011.

[10] D. Harnik, B. Pinkas, and A. Shulmanpeleg, "Side channels in cloud services: Deduplication in cloud storage," *IEEE Security and Privacy*, vol. 8, no. 6, pp. 40–47, 2010.

[11] 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.

[12] M. S. Hwang, C. C. Lee, and T. H. Sun, "Data error locations reported by public auditing in cloud storage service," *Automated Software Engineering*, vol. 21, no. 3, pp. 373–390, 2014.

[13] M. S. Hwang, T. H. Sun, and C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service," *Journal of Circuits, Systems and Computers*, vol. 26, no. 5, 2017.

[14] K. Kim, T. Y. Youn, N. S. Jho, and K. Y. Chang, "Client-side deduplication to enhance security and reduce communication costs," *Etri Journal*, vol. 39, no. 1, pp. 116–123, 2017.

[15] L. Lei, Q. Cai, B. Chen, and J. Lin, "Towards efficient re-encryption for secure client-side deduplication in public clouds," in *International Conference on Information and Communications Security*, pp. 71–84, Nov. 2016.

[16] C. Li, H. Cheung, and C. Yang, "Secure and efficient authentication protocol for power system computer networks," *International Journal of Network Security*, vol. 20, no. 2, pp. 337–344, 2018.

[17] J. Li, J. Li, D. Xie, and Z. Cai, "Secure auditing and deduplicating data in cloud," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2386–2396, 2016.

[18] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing," *International Journal of Network Security*, vol. 18, no. 4, pp. 650–666, 2016.

[19] H. Liu, H. Ning, Y. Zhang, and L. T. Yang, "Aggregated-proofs based privacy-preserving authentication for v2g networks in the smart grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1722–1733, 2012.

[20] X. Liu, W. Sun, W. Lou, Q. Pei, and Y. Zhang, "One-tag checker: Message-locked integrity auditing on encrypted cloud deduplication storage," in *IEEE Conference on Computer Communications*, pp. 1–9, May 2017.

[21] P. Mell and T. Grance, "Draft nist working definition of cloud computing," *National Institute of Standards and Technology*, vol. 53, no. 6, pp. 50–50, 2009.

[22] D. T. Meyer and W. J. Bolosky, "A study of practical deduplication," *ACM Transactions on Storage*, vol. 7, no. 4, pp. 1–20, 2012.

[23] S. Rezaei, M. Ali Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115–131, 2018.

[24] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[25] J. Xiong, Y. Zhang, X. Li, M. Lin, Z. Yao, and G. Liu, "Rse-pow: A role symmetric encryption pow scheme with authorized deduplication for multimedia data," *Mobile Networks and Applications*, vol. 23, no. 3, pp. 650–663, 2018.

[26] J. Xiong, Y. Zhang, L. Lin, J. Shen, X. Li, and M. Lin, "ms-PoSW: A multi-server aided proof of shared ownership scheme for secure deduplication in cloud," *Concurrency and Computation Practice and Experience*, no. 5, 2017.

[27] C. Yang, J. Ren, and J. Ma, "Provable ownership of files in deduplication cloud storage," *Security and Communication Networks*, vol. 8, no. 14, pp. 2457–2468, 2015.

[28] C. Yang, M. Zhang, Q. Jiang, J. Zhang, D. Li, J. Ma, and J. Ren, "Zero knowledge based client side deduplication for encrypted files of secure cloud storage in smart cities," *Pervasive and Mobile Computing*, vol. 41, pp. 243–258, 2017.

[29] J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," in *Communications and Network Security*, pp. 145–153, Oct. 2013.

[30] J. Zhang, P. Li, and M. Xu, "On the security of an mutual verifiable provable data auditing in public cloud storage," *International Journal of Network Security*, vol. 19, no. 4, pp. 605–612, 2017.

Biography

Qianlong Dang is a master degree student in the School of Mathematics and Statistics at Xidian University. His interest focuses on cryptography and network security.

Hua Ma is a professor in the School of Mathematics and Statistics at Xidian University, Xi'an, China. Her research includes security theory and technology in electronic commerce design and analysis of fast public key cryptography theory and technology of network security.

Zhenhua Liu is a professor in the School of Mathematics and Statistics at Xidian University, Xi'an, China. His research interests include public key cryptography, cryptographic theory and security protocols in cloud computing.

Ying Xie is a master degree student in the School of Mathematics and Statistics at Xidian University. Her interest focuses on cryptography and network security.

Design of Key Management Protocols for Internet of Things

Cungang Yang and Celia Li
(Corresponding author: Cungang Yang)
Department of Electrical and Computer Engineering, Ryerson University
350 Victoria Street, Toronto, Canada
(Email: cungang@gmail.com)
(Received June 2, 2019; Revised and Accepted Dec. 16, 2019; First Online Feb. 28, 2020)

Abstract

The Internet of Things (IoT) provides transparent and seamless incorporation of heterogeneous and different end systems. It has been widely used in many applications such as smart homes. However, people may resist the IOT as long as there is no public confidence that it will not cause any serious threats to their privacy. Effective secure key management for things authentication is the prerequisite of security operations. In this paper, we present an interactive key management protocol and a non-interactive key management protocol to minimize the communication cost of the things. The security analysis, numerical analysis and simulation results show that the proposed schemes are efficient and resilient to various types of attacks.

Keywords: IoT Security; Key Management; Ticket Based Authentication

1 Introduction

The Internet of Things (IoT) comprises of billions of devices that can sense, communicate, compute and potentially actuate [1, 3, 4]. IoT involves accessing, monitoring and controlling various sensors and devices over the internet. A great example of the IoT application is smart homes. Household systems like smart smoke-alarms, air quality sensors, smart doorbells, and home monitoring devices can now communicate with smart watches, and activity trackers. After an activity tracker assessed your sleep – determining when you are in light sleep – it can tell your alarm clock to go off. Your alarm clock in unison with your phone will check the weather – just before you wake up (based on your preference and sleep cycle) and tell air conditioners in your car and your home to change the temperature accordingly. Navigation apps on your smart phone – after gathering information from your weather app – can predict how the weather will affect traffic congestion, and plan a route to your work. As the communication between IoT devices may include sensitive and critical data, the security requirements for any IoT-based system are high. To set up a security channel between different devices such as an air quality sensor and a smart watch, a number of security operations (authentication, authorisation, and data integrity) are needed [8]. Since key management is the prerequisite of these security operations, the motivation of this research is thus to develop pairwise key generation and rekeying schemes for IoT devices [6]. Generally, the design of IoT key management protocols has the following security requirements.

1) Secrecy and authenticity: The protocol needs to guarantee that only the intended party learn the key management and that this key is unique and fresh. Security and authenticity need to be protected against attacks such as impersonation, DoS, *etc.* Another security goal is to minimize the negative effects of a comprised key. Keys maybe exposed regardless of the security of the key management protocol that generates them, *e.g.*, by break-ins to a device, poor secure storage for keys, *etc.* Mechanisms like independence between different keys in a system, frequently refreshment, and perfect forward secrecy, as discussed below, address this goal.

2) Key refreshment: The key management protocol must provide automatic mechanism to periodically refresh keys: when a cryptographic key is used actively, the amount of data encrypted with it grows and it becomes easier to perform attacks on the encryption algorithm. To prevent breaking of the security, every key has to be replaced after a time interval.

3) Perfect forward secrecy: Perfect forward secrecy (denoted PFS) refers to the property that disclosure of long-term key does not comprise the session keys from earlier runs. If one encryption key is compromised, only the data encrypted by that specific key is compromised. Some cryptosystems allow session keys to be derived from long term keys, so that if the long term key is compromised, an attacker might

have enough information to figure out session keys and/or decrypt data encrypted using those keys.

4) Key Separation: Different cryptographic functions should use different and independent keys (namely the exposure of one key should not compromise the other). This applies to different functions used in the key exchange protocol as well as the cryptographic functions applied to data during the subsequently sessions. In particular, one has to careful not to reuse the session key for different functions.

Also, new key management protocols should fit the features of IoT and avoid the challenges such as limited bandwidth and vulnerable to attacks. So far, the research on the secure key issues of the IoT is focused on homogenous and heterogeneous wireless sensor networks. Perrig [7] presented a suite of security protocols optimized for sensor networks that they called 'SPINS'. The suite is built upon two secure building blocks, each performing individual required work: SNEP and TESLA. SNEP offers data confidentiality, authentication, integrity, and freshness, while TESLA offers broadcast data authentication. The TESLA protocol, used on regular networks, is modified as a SPINS for use in resource-constrained wireless sensor networks. Disadvantages of this scheme include TESLA overhead from releasing keys after a certain delay and possible message delay. A non-interactive key management approach is introduced in the article "self-certified keys - concepts and applications" [11]. This scheme allows the computation of a session key in a non-interactive manner. Non-interactive key management protocol involve minimal interaction among the nodes of the network which requires global clock. In a key pre-distribution scheme [12,15,16], some keys are preloaded into each sensor before sensor deployment. After deployment, sensor nodes undergo a discovery process to set up shared keys for secure communications. This scheme ensures to some probability that any two sensor nodes can communicate using a pairwise key. This scheme does not, however, ensure that two nodes always are able to compute a pairwise key to use for secure communication. The key management scheme of 802.11i in WLAN is helpful to develop key management protocols in IoT. However, 802.11i has the following weaknesses:

1) The authentication server (AS) works as key distribution center that may not be reachable.

2) More communication costs on the network due to the involvement of the AS.

3) Single point failure of AS.

4) 802.11s does not support Perfect Forward Secrecy. If the primary master key (PMK) is exposed, the session keys will be compromised.

5) The 4-way handshake is vulnerable to DOS attack.

Our work is based on the key management scheme of 802.11i. The motivation of our work is trying to enhance 802.11i with new interactive and non-interactive key management protocols whose design should be able to fit the features of IoT and solve the weaknesses of the key management scheme in 802.11i. The contribution of this paper is developing pairwise key generation and rekey schemes for IoT devices. In particular, we bring in a novel interactive key management protocol which is resilient to attacks and save communication cost. Moreover, we propose a secure non-interactive key management protocol which further reduces the communication cost close to zero. The rest of the paper is organized as follows. Section 2 presents our proposed interactive key management scheme. The non-interactive key management scheme is explained in Section 3. The numerical analysis on the performance of the interactive and non-interactive key management schemes are explained in Section 4. Finally, Section 5 concludes the paper.

2 An Interactive Key Management Scheme

The interactive key management scheme between device A and device S is comprised of two phases that is shown in Figure 1. Notations used in the rest of the paper is summarized in Table 1. In Phase 1, A requests to communicate with S. They mutually authenticate each other with a Ticket-based authentication protocol and generate a Pairwise Master Secret (PMK). In Phase 2, following the establishment of the PMK, a session key rekey protocol is executed to confirm the existence of the PMK and the liveliness of the peers; the session key rekey protocol is resilient to DoS attack and supports Perfect Forward Secrecy (denoted PFS) which refers to the property that disclosure of long-term PMK does not comprise the session keys from earlier runs.

Table 1: Notations

Notation	Description
I_x	ID of node X
P_x	Public key of x
T_x	Ticket issued to x
T_{exp}	Expiry date of a ticket
N_x	Nonce of node x
Sig_x	Digital signature of node x
D_x	Domain name of x
$E_{pub_A}(m)$	Encrypt m using A's public key
V_k	Message authentication code resulting from the application of a MAC key k on a message m

2.1 Phase 1: Ticket-based Authentication and PMK Generation

Tickets are used to establish the trust relationships among entities. For example, devise A will trust devise S if the ticket of S is valid and issued by the ticket agent it trusts. A ticket agent is defined as an authority who issues and manages various types of tickets and can be trusted by various entities in IoT. Before deployment of IoT devices, the network operator, denoted by OP, requests tickets from a ticket agent, one per device, and preinstall the ticket for each node. The OP is also responsible for requesting and distributing new tickets before the current tickets expire.

Following is the structure of a ticket for device R:

$$T_R = \{I_R, I_A, T_{exp}, P_R, D_R, Sig_A\}$$

- T_R: Ticket issued by ticket agent I_A.
- I_R: ID number of the device R that is given this ticket.
- I_A: ID number of the ticket agent who issued ticket T_R to I_R.
- T_{exp}: Expiry date and time of ticket T_R.
- P_R: Public key of I_R, which is used to verify the signature of messages sent by I_R.
- D_R : Domain name of the network that the device is located.
- Sig_A: Digital signature of ticket agent I_A.

With the design of tickets in the design of the key management protocol, the key generation and negotiation of IoT devises do not need the involvement of the third party, such as the key distributed center or authentication server. The messages exchange only between the pair of devices dramatically reduce the communication cost of the network. Following is the messages to be exchanged according to the order of the protocol as shown in Phase 1 of Figure 1.

1) Device A broadcasts its ticket periodically. This message allows device S to detect its presence in order to join the negotiation process. S verifies the digital signature of the ticket agent who issued A's ticket T_A using the ticket agent's public key. We assume that the tickets of all nodes are issued by the same ticket agent and the public key of the agent has been preinstalled in each node. S verifies the domain name of the ticket and ensure that the device it associated is from the same network. S also verifies other information in the ticket such as the ID of the ticket agent and the ticket expiry date.

2) If the above verifications are successful, S extracts A's public key from T_A and generates a message MS which contains S's ticket T_S and two nonce N_{S1} and N_{S2}. S then encrypts the message using the A's public key and sends the encrypted message to the A. Upon receiving the message, A decrypts it using its private key, and verifies the digital signature of the ticket agent who issued the ticket T_S. A verifies the domain name of the ticket and ensure that the device it associated is from the same network. A then verifies other information recorded in ticket T_S such as the ID of the ticket agent who issued T_S and the ticket expiry date.

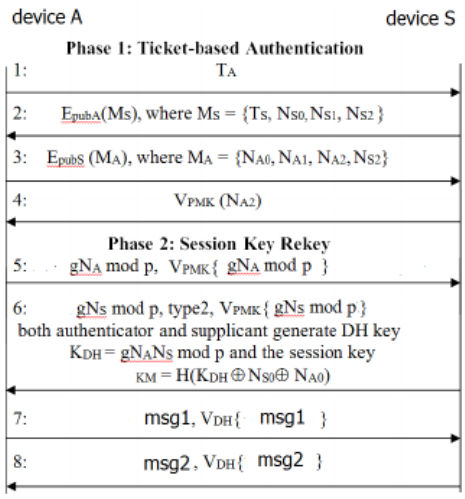


Figure 1: Interactive key management protocol

3) If the above verifications succeed, A retrieves S's public key from ticket T_S, and generates a message MA containing random numbers N_{A1}, N_{A2} and N_{S2}. A then encrypts message MA using S's public key, and sends the encrypted message to S. S will decrypt the message using its private key to retrieve N_{A1}, N_{A2} and N_{S2}. A authenticates S if N_{S2} is correct. Both devices A and S then calculate their shared PMK by applying a hash function H to the message $N_{S1}||N_{A1}$, and N_{S1} and N_{A1} are the random numbers generated in Steps 2 and 3 above. That is, PMK = H($N_{S1}||N_{A1}$).

4) S then uses the key PMK and applies a (predetermined) MAC algorithm on N_{A2} to produce a message authentication code $V_{PMK}(N_{A2})$, which S then sends to A. Upon receiving this message authentication code, A performs the same computation as S just did to produce a message authentication code $V'_{PMK}(N_{A2})$. If $V'_{PMK}(N_{A2}) = V_{PMK}(N_{A2})$, then A has successfully authenticated S, because only S has the knowledge of the shared key PMK and N_{A2} .

In Phase 1, device A and S exchange their tickets and verify the validity of each other's tickets. The trust relationship between A and S from the same network is based on their exchanged tickets which should be issued by a same ticket agent. The results of the protocol are mutual authentication of the pair and the generation of

a shared PMK key which is the basis for the following process to create the session key for data confidentiality.

2.2 Session Key Rekey

The session key rekey protocol is shown in Phase 2 of Figure 1. Here, we assume g and p are public information known by both A and S.

1) In the first message, $g^{N_A} modp$, $V_{PMK} g^{N_A} modp$. Device A generates a random number N_A and calculate the MAC value of g^{N_A} mod p with the PMK key. Device S authenticates A.

2) S generates a random number N_S, $g^{N_S} modp$ and calculate the MAC value of $g^{N_S} modp$ with the PMK key. With this step, A authenticates S. Both A and S then calculate DH key $K_{DH} = g^{N_A N_S} modp$ and their shared session Key KM by applying a hash function H to the message $K_{DH} \oplus N_{S0} \oplus N_{A0}$ where N_{S0} and N_{A0} are the random numbers generated in Steps (1) and (2). That is, $KM = H(K_{DH} \oplus N_{S0} \oplus N_{A0})$.

3) A sends an acknowledgement message, msg1, $V_{KM} msg1$, to S. S authenticates A.

4) S sends an acknowledgement message, msg2, $V_{KM} msg2$, to A. A authenticates S.

The main reason of the DoS attack on the original 4-way handshake of 802.11i is due to the plaintext of message 1. In the new session key rekey protocol, we have generated a shared key to protect the first message so as to avoid blocking and the legitimate authenticator and the supplicant is not necessary to allocate memory to store all the received nonces and the derived PTKs. The interactive key management protocol is resilient to DoS. First, the attacker cannot impersonate device A and forge message 1 since he does not know the PMK and cannot generate the proper MAC value. Any change in the original message 1 cannot be successfully verified by S. Second, the PTK inconsistency in 802.11i 4-way handshake will not happy in the proposed interactive key management protocol. The nonce values of DH key $KDH = g^{N_A N_S} modp$ and the session key, $H(K_{DH} \oplus N_{S0} \oplus N_{A0})$ are all secret values. They both hide from the attackers. Without the knowledge of N_A, N_S, N_{S0}, and N_{A0}, the attacker is not possible to modify the session key or DH key. Thus, the session key inconsistency problem occurred in 802.11i 4-way handshake will not occur in our proposed interactive key management protocol. We consider PMK as the long term key and session key as a short term key. Within the lifetime of PMK, multiple session keys should be updated. our protocol supports PFS. In the scheme, a DH key is introduced and located between the PMK and the session key. PMK key securely transfer the public information g^{NA} mod p and g^{NS}mod p for mutually authenticity of A and S while hide their secret value N_A and N_S accordingly. The knowledge of PMK does not help to derive DH key $g^{N_A N_S}$ mod p because the secret values N_A or N_S are private information of A and S. Even the PMK is exposed, the attacker cannot derive the DH key that is current used, previously used or will be used by valid device A and S. In addition, DH key is the basis to retrieve the session key. For example, the session key is $H(KDH \oplus N_{S0} \oplus N_{A0})$. Hence, the attacker cannot compromise the session keys in case PMK is exposed.

3 The Non-interactive Key Management Protocol (Non-INT)

3.1 Overview

The authenticity of public keys in a public cryptosystem is gained in two different ways: either it is verified by its certificate, or it is verified implicitly during the use of the keys. The latter is introduced by Girault as self-certified keys [2]. Self-certified keys are not verified until it is used for cryptographic function such as signature verification. Public keys of each node are verified without the aid of its public key certificate or an online Certificate Authority (CA) [11]. The concept of self-certified keys is employed in this paper due to its simple non-interactive rekey mechanism. In this section, by coupling the ticket-based technique with the self-certified keys, we obtain a fully non-interactive key management protocol for IoT. In contrast with prior work [11], our techniques for session key update do not require any interaction and do not involve any reliable broadcast communications among devices. Here, we present a new scheme that offers both device A and S to compute or rekey a session key in a non-interactive manner. We achieve this result by using the user-controlled key progression. Compare with interactive key management schemes, the new non-interactive approach further reduce the communication cost of the session key generation and rekey to zero or close to zero.

3.2 Bootstrapping

The network is initialized by the network operator OP. OP chooses large primes p and q with $q|(p-1)$ (q is a prime actor of p-1). OP chooses a random number $K_A \in Zq*$ with order q and generates its (public, private) key pair (y_Z, x_Z). We assume that the public key y_Z, p, q and g are preinstalled to every node of the network. To issue the private key for a device A with identifier ID_A, OP computes the signature parameter $r_A = g^{k_A}(modp)$ and $s_A = x_Z \times h(ID_A, r_A) + k_A(modq)$. r_A is called the guarantee and $x_A = s_A$ is its private key. The public key of A can be computed by any node that has y_Z, ID_A and r_A using the following equation $y_A = y_Z^{h(ID_A, r_A)} \times r_A(modp)$. We denote this initial key pair as $(x_{A,0}, y_{A,0})$. We assume that each node has installed the initial pair of public and private key issued by the OP.

3.3 Self-Certification

The non-interactive key management protocol is comprised of two phases. Phase 1 in Figure 2 is in charge of the PMK key generation and rekey which is interactive. Phase 2 discuss the session key generation and rekey which is non-interactive. For the original non-interactive scheme, for each PMK update, the device A and S need to exchange $r_{A,t} = g^{K_A}, tmodp$ and $r_{S,t} = g^{Ks}, tmodp$ where $1 \leq t \leq n$. This scheme waste valuable bandwidth because each $r_{A,t}$ or $r_{S,t}$ could be as large as 2048 bits or 3072 bits and number n is uncertain since the number of session keys update within a PMK rekey interval is unknown.

Phase 1. Ticket-based authentication and PMK generation.
In Phase 1 of the non-interactive key management protocol.

1) First message T_A includes R and $V_{PMK}R$. Device A generates a random number R and calculate the MAC value of R with the PMK key. Device S authenticates A because only A has the shared PMK to generate the MAC value.
2) Upon receiving the second message, A decrypts it using its private key, and verifies the digital signature of the ticket agent who issued the ticket T_S using the ticket agent's public key. A receives three random numbers N_{S0}, N_{S1}, N_{S2} and $g^{N_S} modp$ where N_S is the secret value generated and hold by S, A verifies other information of ticket T_S such as the ID of the ticket agent who issued T_S and the ticket expiry date.
3) If the above verifications succeed, A retrieves S's public key from ticket T_S, and generates a messageM_A containing $g^{N_A} modp, l, \delta T, F$ and three random numbers $N_{A0}, N_{A1} and N_{A2}$. N_A is the secret value generated and hold by A. A then encrypts message MA using S's public key, and sends the encrypted message to S. S will decrypt the message using its private key and retrieve $g_{N_A} modp$, the length of the one-way hash chain l, session key progression interval δT, lifetime of the PMK F and three random numbers $N_{A0}, N_{A1} and N_{A2}$. Again, S authenticates A in this message.
4) In message 4, S verified A's authenticity. Finally, both A and S calculate the DH key as $K_{DH} = g^{N_A N_S}$ mod p and derive the initial $V_{A,1}$ and $V_{S,1}$ value as $H(K_{DH} \oplus N_{A0} \oplus N_{S0})$. In Phase 1, whenever generate or rekey the PMK, A and S generate their new secret values N_A and N_S which are the basis to derive new session keys in the second phase. After Phase 1, both A and S know their common secret value V as well.

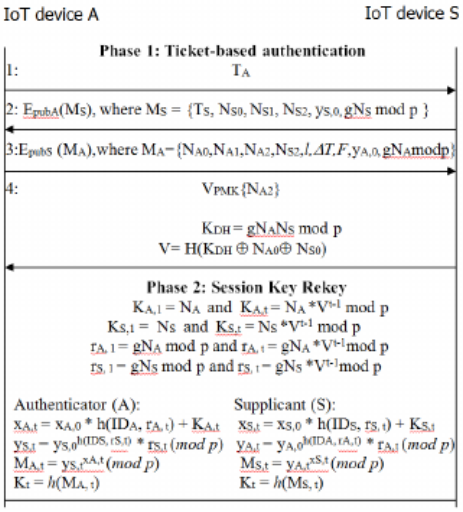


Figure 2: Non-interactive key management protocol

Phase 2. Session key generation and rekey.
$x_{A,0}, x_{S,0}, y_{S,0}$ and $y_{A,0}$ are assigned by the OP. $y_{S,0}$ and $y_{A,0}$ are exchanged by A and S with the second and third messages of Phase 1. We define that

$$\begin{aligned} K_{A,t} &= K_{A,t-1} \times V \bmod p \\ &= K_{A,t-1} \times H(K_{DH} \oplus N_{A0} \oplus N_{S0}) \bmod p \\ &= K_{A,1} \times (H(K_{DH} \oplus N_{A0} \oplus N_{S0})^{t-1} \bmod p \\ K_{A,1} &= N_A. \end{aligned}$$

Thus,

$$\begin{aligned} K_{A,t} &= N_A \times H(K_{DH} \oplus N_{A0} \oplus N_{S0})^{t-1} \bmod p \\ &= N_A \times V_{t-1} \bmod p. \\ r_{A,1} &= g_{A,1}^K \bmod p \\ &= g^{N_A} \bmod p. \\ r_{A,t} &= g_{A,t}^K \bmod p \\ &= g^{N_A} \times (H(K_{DH} \oplus N_{A0} \oplus N_{S0}))^{t-1} \bmod p \\ &= g^{N_A} \times V_{t-1} \bmod p. \end{aligned}$$

For devise S,

$$\begin{aligned} K_{S,1} &= N_S \\ K_{S,t} &= N_S \times V_{t-1} \bmod p, \\ r_{S,1} &= g^{N_S} \bmod p \\ r_{S,t} &= g^S \times V_{t-1} \bmod p. \end{aligned}$$

In Phase 2, A keeps its secret value $K_{A,1} = N_A$ and derives $K_{A,t} = N_A \times V_{t-1} modp$ for the following sessions. S keeps $K_{S,1} = N_S$ and derives $K_{S,t} = N_S \times V_{t-1} modp$ for the following sessions. On the other hand, to derive the public key of the S, A needs to know $r_{S,1}$ and $r_{S,t}$. $r_{S,1} = g^{N_S} modp$ is transferred to A in message 2 of layer 1 while $r_{S,t} = g_{N_S} \times V_{t-1} modp$ can be derived for each session because A know g^{N_S} and V. Each r value we derived will be $\in Zq*$ because q is a prime and all r value are modular p and its value must be in Zq*.

The initial scheme [11] is not a pure non-interactive key management scheme because in their approach the set of $r_{A,t} = g_{V_t} modp$ is shared through message exchange. Compare with the scheme, our protocol allows A and S to generate the $r_{A,t}$ by themselves, and thus no message exchange are involved.

3.4 Security Analysis

For our proposed scheme, the security of the $V_{A,t}$ values depends on the public key algorithm we used in Phase 1 which is safe. The non-interactive has no PFS problem because the PMK has no relationship with the values of $V_{A,t}$ and $V_{S,t}$. If the PMK exposed, it will not compromise the session key.

1) Key security. In the non-interactive key management protocol, the security of the session rekey procedure of Phase 2 depends on the Schnorr signature scheme whose security is based on the intractability of discrete logarithm problems. The Schnorr signature scheme has been provably secure in a random oracle model [5, 13]. To derive the value of the session key, the attacker has to figure out $x_{A,t}$ and $y_{S,t}$.

$$\begin{aligned} x_{A,t} &= x_{A,0} \times h(ID_A, r_{A,t}) + K_{A,t} \\ &= x_{A,0} \times h(ID_A, g^{N_A} \times V_{t-1} modp) \\ &\quad + K_{A,t} \bmod p \\ &= x_{A,0} \times h(ID_A, g^{N_A} \times V_{t-1} modp) \\ &\quad + N_A \times V_{t-1} \bmod p. \\ y_{S,t} &= y_{S,0}^{h(ID_S, r_{S,t})} \times r_{S,t} \bmod p \\ &= y_{S,0}^{h(ID_S, g^{N_S})} \times V_{t-1} \bmod p) \times r_{S,t} \bmod p \\ &= y_{S,0}^{h(ID_S, g^{N_S})} \times V_{t-1} \bmod p) \times g^{N_S} \\ &\quad \times V_{t-1} \bmod p, \end{aligned}$$

where only the ID of A and S, p and g are public known. Other parameters are hiding from the attackers. Thus the session keys cannot be disclosed to attackers.

2) Key refreshment. For the non-interactive key management protocol, the update of PMK is carried out in Phase 1 while the session key rekey is automatically implemented by device A and S. Whenever the session key needs rekeying, the Phase 2 of each protocol will be carried out.

3) Perfect forward secrecy. The only value in Phase 1 relating to the generation of session key is V. $V = H(K_{DH} \oplus N_{A0} \oplus N_{S0})$. If the PMK is exposed, it cannot derive DH key. Thus, we can say that the attacker cannot compromise the session key if PMK is exposed.

4) Key separation.

   a. PMK and Session key: The PFS analysis shows that PMK is independent from the session key. That is, if PMK is exposed, the session key will not be compromised. Due to the same reason, if a session key is exposed, the PMK cannot be compromised either.

   b. PMK and DH key: In the non-interactive key management protocol, DH key $K_{DH} = g^{N^A N^S} modp$, the N^A and N^S are secret random numbers that only known by the authenticator and supplicant. The PMK and session key are independent: if PMK is exposed, it does not help to figure out the DH key. On the other hand, if DH key is exposed, the PMK will not be compromised.

   c. DH and Session key: The session key $K_t = h(MA, t) = y_{S,t}^{x-A,t} modp$. To derive the session key, we have to know $x_{A,t}$ and $y_{S,t}$

$$\begin{aligned} x_{A,t} &= x_{A,0} \times h(ID_A, r_{A,t}) + K_{A,t} \\ &= x_{A,0} \times h(ID_A, g^{N_A} \times V_{t-1} \bmod p) \\ &\quad + K_{A,t} \bmod p \\ &= x_{A,0} \times h(ID_A, g^{N_A} \times V_{t-1} \bmod p) \\ &\quad + N_A \times V_{t-1} \bmod p \\ &= x_{A,0} \times h(ID_A, g^{N_A} \\ &\quad \times h(K_{DH} \oplus N_{A0} \oplus N_{S0})^{t-1} \bmod p) \\ &\quad + N_A \times h(K_{DH} \oplus N_{A0} \oplus N_{S0})^{t-1} \\ &\quad \bmod p \\ y_{S,t} &= y_{S,0}^{h(ID_S, r_{S,t})} \times r_{S,t} \bmod p \\ &= y_{S,0}^{h}(ID_S, g^{N_S} \times V^{t-1} \bmod p) \times r_{S,t} \\ &\quad \bmod p \\ &= y_{S,0}^{h}(ID_S, g^{N_S} \times V^{t-1} \bmod p) \times g^{N_S} \\ &\quad \times V_{t-1} \bmod p \\ &= y_{S,0}^{h}(ID_S, g^{N_S} \\ &\quad \times h(K_{DH} \oplus N_{A0} \oplus N_{S0})^{t-1} \bmod p \\ &\quad \times g_{N_S} \times h(K_{DH} \oplus N_{A0} \oplus N_{S0})^{t-1} \\ &\quad \bmod p. \end{aligned}$$

If DH key is exposed, the session key of non-interactive protocol cannot be compromised since only g, p, KDH and IDs of authenticator and supplicant are know. Other parameters are hiding from the attackers. Due to the same reason, if the session key is exposed, the attacker still cannot derive the DH key.

4 Performance Analysis and Simulation

4.1 Numerical Analysis

We compare our proposed interactive key management protocol (INT), and the non-interactive protocol (Non-INT) with the EAP-TLS and 4-way handshake protocol.

We choose EAP-TLS and 4-way handshake protocol for comparison because EAP-TLS and 4-way handshake is the authentication protocol in IEEE 802.11i. 4-way handshake protocol is vulnerable to DoS attack while our proposed protocols do not have. The performance is measured in terms of Latency of the key generation protocol, which is defined as the summation of the computation cost and communication cost.

- Computation costs, which are the latencies (in milliseconds) incurred by the security operations such as encryption, decryption and hashing [9];
- Communication costs, which indicate the number of messages exchanged between the neighbouring devices to complete an key generation session.

Computation costs. Table 2 lists the security operations, the current state-of-the-art algorithms implementing the operations, and the computation time each of these algorithms incurs. Since the encryption operation of RSA is a modular exponentiation, we assume that the cost of modular exponentiation is the same as that of RSA encryption. The original EAP-TLS and 4-way handshake protocol performs one public-key encryption, one public-key decryption, one signature generation, three signature verifications, five MAC operation and two hash function (assuming that A and S compute the MAC key KMAC in parallel). The fourth column of Table 2 records the above numbers of operations. 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 the EAP-TLS and 4-way handshake protocol performs, we obtain a total computation cost of 97.9645ms, as shown in the third last row of the fourth column.

Similarly, the fifth and sixth columns of Table 3 list the numbers of security operations the proposed INT and non-INT perform, respectively. Applying similar calculations as above, we obtain the computation costs of the proposed INT and non-interactive protocol, which are 108.09ms and 110.94ms, respectively. The Non-INT protocol includes an interactive PMK generation and a non-interactive session key generation. The latency of PMK and session key generation in non-INT protocol includes two times Epub, two times Dpub, two times Vsig, one time MAC and five times modular exponentiation operations. Two devices pre-compute their session keys before the session key is expired. Thus, its computation cost for the latency of session key generation in non-interactive protocol is zero.

Communication costs. For the PMK generation, Table 2 lists the number of messages involved in each of the three protocols we compare. The proposed INT and Non-INT require less messages to be exchanged than EAP-TLS and 4-way handshake. For the session key generation, Table 3 lists the number of messages involved in each of the three protocols we compare. The proposed INT has the same number of messages to be exchanged as EAP-TLS and 4-way handshake. There is no message exchange between the two devices to negotiate session key in the non-interactive protocol, and thus their communication cost is zero. In summary, considering both computation and communication costs, the latency of EAP-TLS, INT and Non-INT are 385.16ms, 327.77ms and 182.74ms, respectively.

4.2 Simulation Results

We further evaluate and compare the performance of EAP-TLS, INT and Non-INT protocols under realistic network settings using simulations. The 600m x 600m network has one home device, which is placed in the center area of the square. We assume a number of neighbouring devices could directly communicate with the home device to illustrate the overhead of the key generation approach used by EAP-TLS, INT and Non-INP. We varied the number of neighbors from 1 to 30. Each data point in the graphs is the average of 10 runs using different random seeds. The graphs are plotted with a confidence interval of 95%. We conducted two experiments as function of:

1) Number of neighboring devices: We measure the key generation latency as function of the number of the neighboring devices. We assume that up to 30 neighboring devices implement the EPH-TLS, INT or Non-INT protocol with the home device simultaneously. We calculate the average key generation delay, averaged over all neighbors participating in the experiment. We also keep track of the maximum key generation delay, the maximum value among all neighbors of the home device. The messages of the key generation protocols may get lost. We measure the success rate of key distribution for 10 neighbors. The success rate is defined as follows: if the home device has m neighbors and we consider eight messages of INT as an example, the number of key generation messages for all neighbor's key generation request is m * 8. Assume each experiment run 10 times with different seeds, the total messages regarding to a client's request is 10 * m * 8. If the simulation result shows that s messages are lost, the success rate of m neighbors is (10 * m * 8 - s)/ (10 * m * 8).

2) Background traffic load: We calculate the average and maximum key generation latency of 10, 20 and 30 neighbors as a function of background traffic. The data rate for both scenarios is varied from 10 Mbits/s to 50 Mbits/s. Data rate is 0 means that there is no background traffic. We also measure the success rate of key generation messages as the function of background traffic. The data rates various from 10Mbits/s to 50MBits/s. Here, we assume the home

Table 2: Cost of PMK

Operations	Algorithm	Time(ms)	EAP-TLS	INT	Non-INT
Epub	RSA	1.42	1	2	2
Dpub	RSA	33.3	1	2	2
Gsig	ECDSA	11.6	1	0	0
Vsig	ECDSA	17.2	3	2	2
MAC	HMAC	0.0015	5	5	1
Hashing	SHA-1	0.009	2	1	0
Modular Exponentiation		1.42	0	3	5
Total computational cost			97.9645	108.09	110.94
# of messages			12	8	4
Latency of PMK key			313.36	251.69	182.74

Table 3: Cost of session key generation

Operations	Algorithm	Time(ms)	4-way Handshake	INT	Non-INT
MAC	HMAC	0.0015	3	4	0
Hashing	SHA-1	0.009	0	1	0
Modular Exponentiation		1.42	0	3	0
Total computational cost			0.0045	4.275	0
# of messages			4	4	0
Latency of session key			71.8	76.08	0

device has 10 neighbors. Following is a detailed discussion of the experimental results.

Experiment 1. Function of number of neighboring BMAPs. The graph in Figure 3 and Figure 4 show the average latency and maximum latency as function of the home BMAP's neighbors. As the number of neighboring devices increases from 1 to 30, the average latency of EAP-TLS, INT and Non-INT increases as expected, by approximately 69.7%, 81.1% and 76.3% respectively. The maximum latency of the protocols increase by approximately 93.1%, 89.5% and 98.7%. More clients imply more key distribution requests to be processed by the home device, and more channel contention around the home device, resulting in longer delay Figure 5 shows the success rate as the function of neighbors. According to the formula we provided in section IV, the success rate of key distribution messages of 10 neighboring devices in EAP-TLS, INT and Non-INT are at the range of 98.3% and 99.6%. We observe that the number of neighboring devices does not have a big impact on its success rate, which is a positive attribute of the key generation scheme.

Experiment 2. Function of background traffic load We examine how background traffic may affect the average latency and maximum latency if 10 neighboring devices request key generation

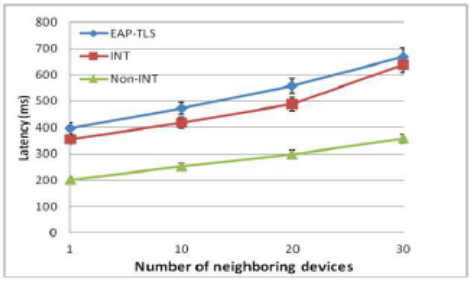


Figure 3: Average latency as function of number of neighbors

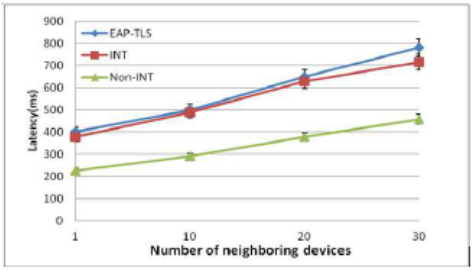


Figure 4: Maximum latency as function of number of neighbors

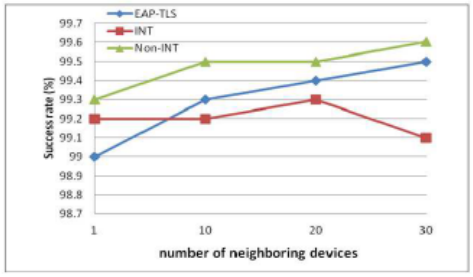


Figure 5: Success rate as function of number of neighbors

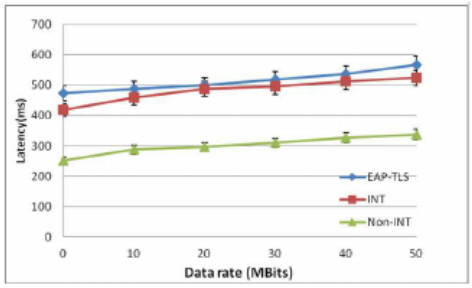


Figure 6: Average latency as function of background traffic

from the home device. Figure 6 shows average 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, average latency of neighboring devices is enlarged. Higher data rate implies more background traffic to be processed by the home device, and more channel contention around the home device, resulting in longer delay. Figure 7 shows the maximum latency of 10 neighboring devices. As the data rate increases from 0 to 50MBits/s, the maximum latency of EAP-TLS, INT and Non-INT increases as expected, by approximately 34.3%, 19.9% and 18.2% respectively.

Figure 8 shows the success rate as the function of data rate. The success rate of key generation messages of 10 neighboring devices is at the range of 96.8% and 99.7%. We observe the success rate is higher if there is no background traffic (data rate is 0). However, the data rate does not have a big impact on success rate, which is a positive attribute of the key generation scheme.

5 Conclusion

Security has become the central issue for IoT and key management plays a critical role to ensure data confidentiality and integrity. A new design of ticket-based authentication protocol, an interactive key management protocol and a non-interactive key management protocol en-

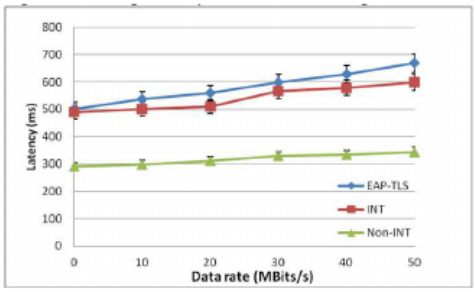


Figure 7: Maximum latency as function of background traffic

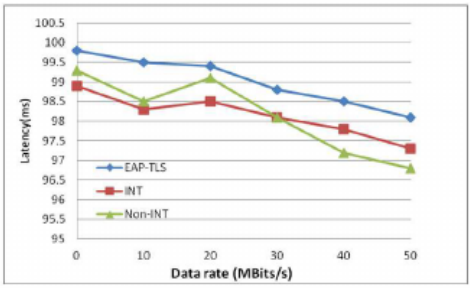


Figure 8: Success rate as function of background traffic

hanced the security of 4-way handshake and at the same time have lower latency than that of EAP-TLS and 4-way handshake. Unlike EAP-TLS and 4-way handshake. the interactive key management protocols support PFS and resists DoS attack,

References

[1] M. Frustaci, P. Pace, G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet of Things Journal*, vol. 5, no. 4, 2018.

[2] M. Girault, "Self-certified public keys," in *Workshop on the Theory and Application of of Cryptographic Techniques*, vol. 547, pp. 490-497, 2001.

[3] L. Gutiérrez-Madroñal, M. F. Wagner, I. Medina-Bulo, "Test event generation for a fall-detection IoT system," *IEEE Internet of Things Journal*, vol. 6, no. 4, 2019.

[4] M. S. Henriques and N. K. Vernekar, "Using symmetric and asymmetric cryptography to secure communication between devices in IoT," in *International Conference on IoT and Application*, May 2017. DOI:10.1109/ICIOTA.2017.8073643.

[5] P. Horster, M. Michels, H. Peterson, "Meta-ElGamal signature schemes," in *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, pp. 96-107, 1994.

[6] 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 Inno-*

vative Computing, Information and Control, vol. 5, no. 8, pp. 2107-2124, Aug. 2009.

[7] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 52-61, 2003.

[8] L. Liu, Z. Cao, O. Markowitch, "A note on design flaws in one aggregated-proof based hierarchical authentication scheme for the internet of things," *International Journal of Electronics and Information Engineering*, vol. 5, no. 2, pp. 88–92, 2016.

[9] 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.

[10] D. Manz, J. Alves-Foss and S. Zheng, "Network simulation of group key management protocols," *Journal of Information Assurance and Security*, pp. 67-79, 2008.

[11] H. Petersen, P. Horster, "Self-certified keys - concepts and applications," in *Communications and Multimedia Security*, pp. 102-116, 1997.

[12] M. Saikia, Md. A. Hussain, "Combinatorial group based approach for key pre-distribution scheme in wireless sensor network," in *International Conference on Computing, Communication and Automation (ICCCA'17)*, 2017. DOI: 10.1109/CCAA.2017.8229851.

[13] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp.161-174, 1994.

[14] S. Vanstone, *Deployments of Elliptic Curve Cryptography*, 2005. (`http://www.cacr.math.uwaterloo.ca/conferences/2005/ecc2005/vanstone.pdf`)

[15] O. Yagan, A. M. Makowski, "Wireless sensor networks under the random pairwise key predistribution scheme: Can resiliency be achieved with small key rings?," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, 2016.

[16] J. Zhao, "Probabilistic key predistribution in mobile networks resilient to node-capture attacks," *IEEE Transactions on Information Theory*, vol. 63, no. 10, 2017.

Biography

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.

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.

Research on Medical Image Encryption Method Based on Improved Krill Herb Algorithm and Chaotic Systems

Jing Bi, Shoulin Yin, Hang Li, Lin Teng, and Chu Zhao

(Corresponding author: Shoulin Yin and Hang Li)

Software College, Shenyang Normal University
253 Huanghe N St, Huanggu Qu, Shenyang Shi, Liaoning Sheng, China
(Email: yslinhit, lihangsoft@163.com)

(Received Nov. 30, 2018; Revised and Accepted June 25, 2019; First Online Sept. 21, 2019)

Abstract

Based on chaotic system and improved krill herb algorithm, so password flow is generated, and we put forward an effective medical image encryption method in this paper. The new method adopts adaptive function to sort krill individuals and select the best results. The output sequence of sub-key formed by location of part of individual encrypts medical image. Experimental results show that the new algorithm is more complexity and randomness than ordinary pseudo-random sequence generator. It limits the possibility of inferring all keys by attacker, and ensures the effectiveness of resistance in known plaintext attack. Finally, it achieves high security of encryption for medical images compared with other encryption methods.

Keywords: Chaotic System; Krill Herb Algorithm; Medical Image Encryption

1 Introduction

Medical image security [7, 17, 25] technology plays an important role in the military, medical fields and other highly confidential fields. In the transmission or archiving of encrypted images, it is necessary to analyze them with keys in the encryption stage. In particular, data compression and authentication of reversible data hiding in the encryption field must be completed in the encryption stage. For example, in a cloud computing scenario, if the original content of the image or the key used to encrypt the image is not known, the secret message can be encapsulated in the encrypted image. But in the decoding stage, the original image must be completely recoverable and the secret information must be extracted without errors. Therefore, there is a trade-off between packaging capacity and reconstructed image quality. Compared with text data, image data has larger amount of encrypted data and redundant information and stronger resistance to malicious exhaustive attack [8–11].

Compared with the traditional encryption system, chaotic system [22, 23] has the advantages of stronger. The trajectory of the aperiodic and extreme sensitivity to initial conditions, nonlinear, each state ergodicity, unpredictability and other features, are attached great importance by many scholars and experts. Krill herb optimization (KH) algorithm [18, 19] is a bionic algorithm macro sense, it mimics all life and the generation of intelligence and evolution process. It is the optimal random search algorithm based on natural selection principle, which has a simple and good robustness, parallel and adaptive nature, *etc.*

In recent years, it has been widely used in cryptography, machine learning, neural network training, combinatorial optimization and other fields due to its great potential in solving complex optimization problems [20, 21, 24]. The application of krill swarm algorithm for encrypting data is also one of the research frontiers in medical image encryption field currently.

Cao [4] presented a medical image encryption algorithm using edge maps derived from a source image. The algorithm was composed by three parts: bit-plane decomposition, generator of random sequence, and permutation. Chen [5] proposed an adaptive medical image encryption algorithm based on improved chaotic mapping in order to overcome the defects of the existing chaotic image encryption algorithm. First, the algorithm used Logistic-sine chaos mapping to scramble the plain image. Then, the scrambled image was divided into 2-by-2 sub blocks. By using the hyper-chaotic system, the sub blocks were adaptively encrypted until all the sub block encryption was completed. Nematzadeh [15] aimed at proposing a medical image encryption method based on a hybrid model of the modified genetic algorithm (MGA) and coupled map lattices. First, the proposed method employed coupled map lattice to generate the number of secure cipher-images as initial population of MGA. Next, it applied the MGA to both increase the entropy of the cipher-

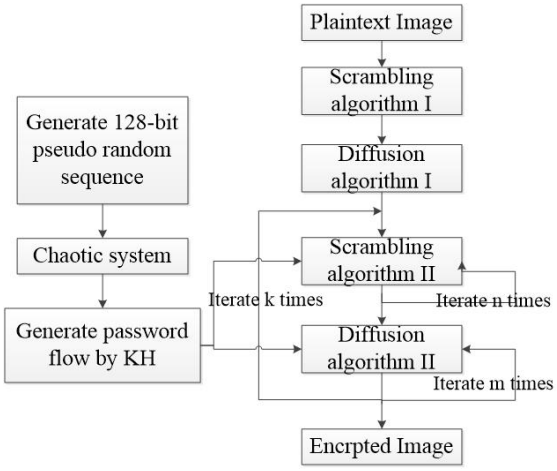


Figure 1: Proposed encryption framework

images and decreased the algorithm computational time. Prabhavathi [16] proposed a chaos-based image encryption conspire utilizing Lorentz map and Logistic condition with numerous levels of diffusion. The Lorentz map was utilized for confusion and the Logistic condition for diffusion, in this work, stenography and encryption systems were joined to ensure the patient secrecy, and increment the security in medicinal images. However, the encryption efficiency is not high. To perfect these demerits, we present an improved krill herb algorithm and chaotic system for medical image encryption.

The rest of this paper is organized as follows. Section 2 introduces the framework of proposed image encryption algorithm. Section 3 provides improved krill herb algorithm for image encryption. Section 4 provides chaotic encryption. Section 5 shows the results and discussions. The conclusions are drawn in the final section.

2 Framework of Proposed Medical Image Encryption

As shown in Figure 1, the krill herb algorithm is used to generate the cryptographic stream, which mainly includes three steps:

1) Firstly, the pseudo-random number generator is used to generate the 128-bit random sequence and the krill herb algorithm is used to generate the cryptographic stream;

2) The plaintext image is scrambled and diffused to obtain the scrambled graph by a series of algorithms;

3) Combined with the password stream generated by krill herb algorithm, the scrambled graph is encrypted again to obtain the final encrypted image.

3 Improved Krill Herb Algorithm for Medical Image Encryption

The KH algorithm is a new heuristic intelligent optimization algorithm, which is mainly based on the simulation study of the survival process of the Antarctic krill group in the marine environment. For each krill individual, its location update is mainly affected by three factors:

1) Induced exercise (induction of surrounding krill);

2) Foraging activities;

3) Random diffusion.

The speed update formula for krill individuals uses the following Lagrangian model:

$$\frac{dx_i}{dt} = N_i + F_i + D_i.$$

Where, N_i, F_i, D_i represent induced movement, foraging movement and random diffusion, respectively.

The formula for the three factors is constructed as follows:

$$\begin{aligned} N_i &= N^{max}\alpha_i + w_n N_i^{old} && (1)\\ F_i &= V_f\beta_i + w_f F_i^{old} && (2)\\ D_i &= D^{max}(1 - \frac{t}{t_{max}}) \cdot \delta. \end{aligned}$$

Where, N^{max}, V_f and D_{max} represent the maximum induction speed, the maximum foraging speed and the maximum diffusion speed, respectively. α_i, β_i, δ represent the direction of induction, the direction of foraging and the direction of diffusion, respectively. w_n and w_f denote the induced weight and the foraging weight respectively. t and t_{max} are the current iteration number and the maximum number of iterations.

The position update formula for krill individuals in the interval t to $t + \Delta t$ is as follows:

$$\begin{aligned} x_i(t + \Delta t) &= x_i(t) + \frac{dx_i}{dt}(\Delta t).\\ \Delta t &= C_t \sum_{j=1}^{NV}(UB_j - LB_j). \end{aligned}$$

Where Δt is the scaling factor of the velocity vector, C_t is the step size scaling factor, taking a constant between [0,2]. NV represents the number of variables. UB_j and LB_j are the upper and lower bounds of the $j-th$ variable, respectively.

In order to further improve the performance of the algorithm, the genetic operator (crossover or mutation) is executed in the algorithm. After testing, the crossover operator is more effective.

$$\begin{aligned} x_{i,m} &= x_{r,m} \quad rand_{i,m} < C_r\\ x_{i,m} &= x_{gbest,m} + \mu(x_{p,m} - x_{q,m}) \quad rand_{i,m} < Mu. \end{aligned}$$

Where C_r is a crossover operator, M is a genetic operator, and $rand$ is an uniformly distributed random number on [0,1]. u is a constant in [0,1].

In the KH algorithm, assuming that $\alpha_i = \beta_i = 0$ in Equations (1) and (2), the krill individuals will always induce movement and foraging movement with $w_n N_i^{old}$ and $w_f F_i^{old}$ until the boundary. It can be seen that the larger w_n and w_f are beneficial to jump out of the local minimum point, and the algorithm has strong global search ability; the smaller w_n and w_f are beneficial to the accurate local search of the current region, and improve the local search ability of the algorithm. Therefore, reasonable adjustment of the induced weight w_n and foraging weight w_f is the key to efficient algorithm search and avoid falling into local optimum. This paper proposes a time-based nonlinear diminishing strategy, namely:

$$w_n = w_f = \frac{w_{max} - w_{min}}{t_{max}} \cdot (t_{max} - t) + w_{min} \cdot rand.$$

Where t and t_{max} are the current iteration number and the maximum number of iterations, respectively. w_{max} and w_{min} represent the maximum and minimum values of the induced weight and the foraging weight, respectively. This strategy makes the overall w_n and w_f of the algorithm gradually decrease. The introduction of the random number rand changes its monotonic mode of linear decrement, so that the algorithm can adapt to the current search situation well throughout the iterative process, thus more effectively adjusting the global search and local exploration ability of the algorithm.

In the KH algorithm, the krill individuals are randomly distributed in various locations in the solution space, and the position of the food is calculated based on the current location of the krill individuals. However, as the iterative process progresses, the location of the krill population and the location of the food tend to be the same, so that the exchange of information between the krill population and the exchange of information between the krill population and the food location becomes significant. Getting smaller and smaller. So we add random disturbances when generating a new generation of populations, and update the formula as follows:

$$x_i(t + \Delta t) = x_i(t) + \frac{dx_i}{dt} \cdot (\Delta t) \cdot rand.$$

Through the random perturbation of the above update method, the amount of information contained in the new generation of krill group can be increased, so that the krill individual who falls into the local optimum jumps out of the local optimum and moves toward the global optimal direction. In the later stage of the algorithm, the local exploration ability of the algorithm can be obviously enhanced and the accuracy of the solution can be improved.

4 Chaotic Encryption

Firstly, Logistic chaotic mapping was used to scramble the pixels of the original image [3,6,14]. Logistic chaotic mapping expression is $x_{n+1} = \mu \cdot x_n(1-x_n)$, where $x_n \in (0,1)$, $n = 1, 2, \cdots$. $0 < \mu \leq 4$ is a bifurcation parameter. $x_i \in (0,1)$, $i = 0, 1, 2, \cdots$. When $\mu \in [3.6, 4]$, Logistic systems are chaotic systems. When the initial value is x_0, the sequence can be obtained. The detailed steps are as follows:

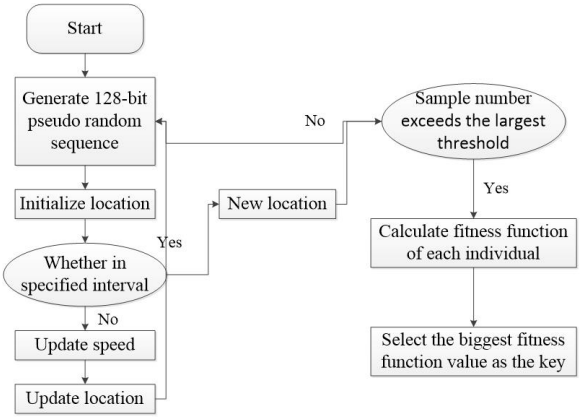


Figure 2: The process of generating a cryptographic stream using KH algorithm

1) Supposing the original image size is $m \times n$, the initial value of chaotic mapping is μ and x_i to generate one-dimensional chaotic sequence $x_k, k = 1, 2, \cdots, m \times n$.

2) The elements in x_k are rearranged with a certain rule, that is, the scrambling operation, and then the new sequence is obtained, denoted as x'_k.

3) The image obtained from the previous step of scrambling is evenly divided into two sub-blocks, denoted as P1 and P2. The XOR operation of P1 and P2 is performed to obtain P11, and the XOR operation of P11 and P2 is performed to obtain P22. Finally, a new matrix is synthesized from P11 and P22 to complete the image diffusion.

4) Repeat the above steps for n scrambling and m diffusion.

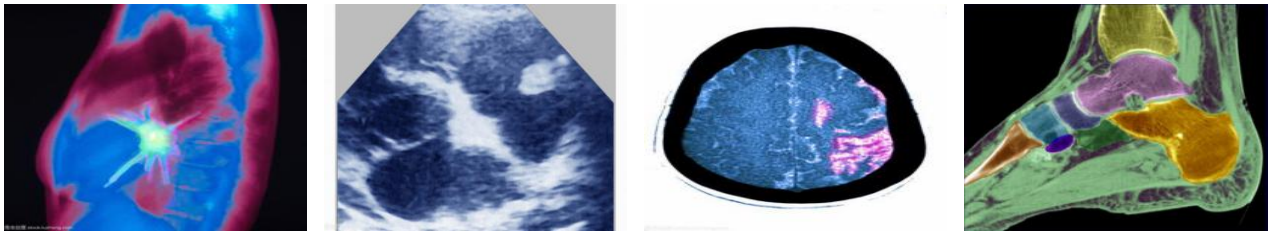
5 Experiment Results

We conduct experiments on matlab. The image size is 128×128pixel. Figure 3 is the original medical image. Figure 4 is the corresponding encrypted image. Figure 5 is the decrypted image. Figure 6 is the gray histogram of Figure 3. Figure 7 is the gray histogram of Figure 4.

According to Figure 4, it can be obtained that:

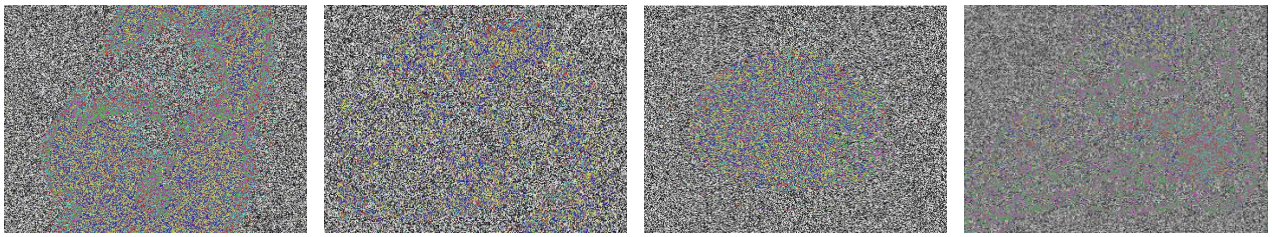
1) The encrypted image approximates a white noise;

2) The decrypted image is relatively clear, and the pixels are almost close to the original image;



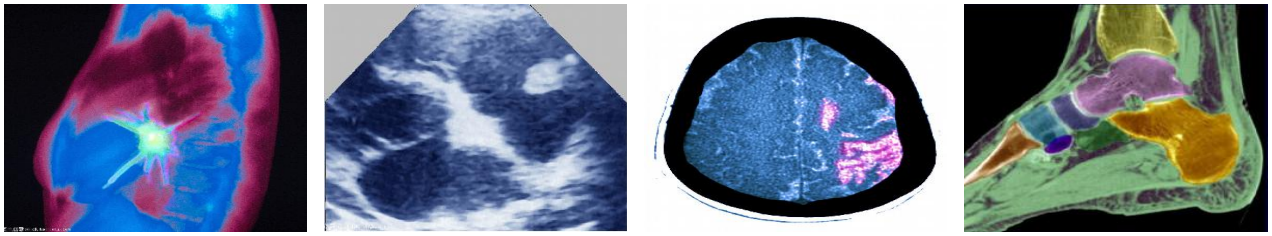
(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

Figure 3: Testing images



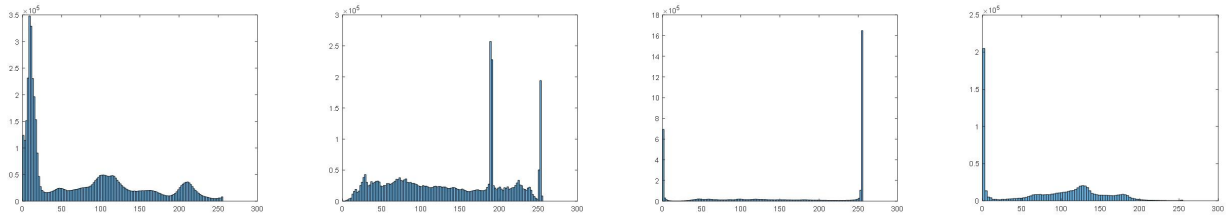
(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

Figure 4: Encrypted images



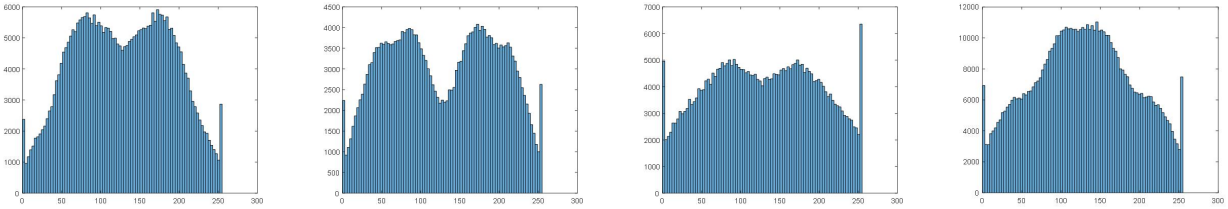
(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

Figure 5: Decrypted images



(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

Figure 6: Gray histogram



(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

Figure 7: Gray histogram

3) The histogram of the encrypted image is smooth, which is different from the histogram of the original image, indicating the effectiveness of the image encryption algorithm based on KH algorithm and chaotic system.

We also make comparison with some newest encryption methods including DHDL [12], EST [13], SDWT [1] and PSC [2]. Operational efficiency analysis is shown in Table 1.

Table 1: Time analysis / s

Method	DHDL	EST	SDWT	PSC	Proposed
Image 1	0.38	0.26	0.25	0.31	0.12
Image 2	0.58	0.46	0.31	0.47	0.23
Image 3	0.37	0.28	0.22	0.23	0.14
Image 4	0.41	0.37	0.26	0.22	0.17

The following two factors are tested for demonstrating the efficient of proposed method in Tables 2 and 3.

1) Pixel change rate:

$$N_{PC} = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} d(i,j)}{m \times n} \times 100\%,$$

where $d(i,j) = 1$, if $p(i,j) = p'(i,j)$.

2) The mean intensity varies uniformly:

$$U = \frac{100}{m \times n} \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} |\frac{p(i,j) - p'(i,j)|}{255}$$

Table 2: Pixel change rate comparison

Method	DHDL	EST	SDWT	PSC	Proposed
Image 1	85.62	88.46	89.74	91.59	98.67
Image 2	89.67	91.06	92.38	92.97	98.75
Image 3	86.79	93.65	95.62	96.83	99.12
Image 4	90.32	92.65	96.37	96.22	99.14

Table 3: U comparison

Method	DHDL	EST	SDWT	PSC	Proposed
Image 1	26.34	28.54	30.94	29.67	32.44
Image 2	25.97	28.66	30.12	31.12	33.09
Image 3	26.11	26.97	30.58	28.66	32.95
Image 4	26.21	27.56	31.68	30.38	34.25

The tables show that the new method has better encryption effect than other three methods.

6 Conclusions

In this paper, we put forward a new medical image encryption algorithm based on KH algorithm and chaotic system. The algorithm by using the improved KH algorithm has a simple and good robustness, nature parallel and adaptive advantages. It effectively combines with chaotic system for image encryption. To test and verify the effectiveness of the proposed algorithm, the experimental results show that the new method has good effect of encryption.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. K. Abdmouleh, A. Khalfallah, M. S. Bouhlel, "A novel selective encryption DWT-based algorithm for medical images," in *International Conference on Computer Graphics, Imaging and Visualization*, pp. 79-84, 2017.

[2] M. Ahmad, M. Z. Alam, Z. Umayya, *et al.*, "An image encryption approach using particle swarm optimization and chaotic map," *International Journal of Information Technology*, vol. 10, no. 3, pp. 247-255, 2018.

[3] Ashish, J. Cao, R. Chugh, "Chaotic behavior of logistic map in superior orbit and an improved chaos-based traffic control model," *Nonlinear Dynamics*, vol. 94, no. 2, pp. 959-975, 2018.

[4] W. Cao, Y. Zhou, C. L. P. Chen, *et al.*, "Medical image encryption using edge maps," *Signal Processing*, vol. 132, pp. 96-109, 2017.

[5] X. Chen, C. J. Hu, "Adaptive medical image encryption algorithm based on multiple chaotic mapping," *Saudi Journal of Biological Sciences*, vol. 24, no. 8, pp. 1821, 2017.

[6] O. S. Faragallah, "Optical double color image encryption scheme in the Fresnel-based Hartley domain using Arnold transform and chaotic logistic adjusted sine phase masks," *Optical & Quantum Electronics*, vol. 50, no. 3, pp. 118, 2018.

[7] S. Haddad, G. Coatrieux, M. Cozic, *et al.*, "Joint watermarking and lossless JPEG-LS compression for medical image security," in *International Conference on Watermarking and Image Processing*, vol. 38, no. 4, pp. 198-206, 2017.

[8] L. C. Huang, M. S. Hwang, L. Y. Tseng, "Reversible and high-capacity data hiding in high quality medical images," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 1, pp. 132–148, 2013.

[9] L. C. Huang, M. S. Hwang, and L. Y. Tseng, "Reversible data hiding for medical images in cloud computing environments based on chaotic Henon map,"

Journal of Electronic Science and Technology, vol. 11, no. 2, pp. 230–236, 2013.

[10] L. C. Huang, L. Y. Tseng, M. S. Hwang, "A reversible data hiding method by histogram shifting in high quality medical images," *Journal of Systems and Software*, vol. 86, no. 3, pp. 716-727, Mar. 2013.

[11] L. C. Huang, L. Y. Tseng, and M. S. Hwang, "The study on data hiding in medical images," *International Journal of Network Security*, vol. 14, no. 6, pp. 301-309, Nov. 2012.

[12] S. M. Ismail, L. A. Said, A. A. Rezk, *et al.*, "Image encryption based on double-humped and delayed logistic maps for biomedical applications," in *International Conference on Modern Circuits and Systems Technologies*, pp. 1-4, 2017.

[13] T. Jiang, K. Zhang, J. Tang, "Securing medical images for mobile health systems using a combined approach of encryption and steganography," in *International Conference on Intelligent Computing*, pp. 532-543, 2018.

[14] X. Lou, W. Tang, X. Chen, "A high capacity quantum weak blind signature based on logistic chaotic maps," *Quantum Information Processing*, vol. 17, no. 10, pp. 251, 2018.

[15] H. Nematzadeh, R. Enayatifar, H. Motameni,*et al.*, "Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices," *Optics & Lasers in Engineering*, vol. 110, pp. 24-32, 2018.

[16] K. Prabhavathi, C. P. Sathisha, K. M. Ravikumar, "Region of interest based selective medical image encryption using multi Chaotic system," in *International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques*, 2018. DOI: 10.1109/ICEECCOT.2017.8284614.

[17] A. Rai, H. V. Singh, "SVM based robust watermarking for enhanced medical image security," *Multimedia Tools & Applications*, vol. 76, no. 18, pp. 1-14, 2017.

[18] S. Singh, S. Tripathi, N. Kumar, "An enhanced security-aware dynamic packet scheduling scheme for wireless networks using intelligent time slice-based krill herd algorithm," *Journal of Electromagnetic Waves & Applications*, vol. 32, no. 16, pp. 1-22, 2018.

[19] Y. Sun, S. Yin, J. Liu, "Novel DV-hop method based on krill swarm algorithm used for wireless sensor network localization," *Telkomnika Telecommunication, Computing, Electronics and Control*, vol. 14, no. 4, pp. 1438, 2016.

[20] L. Teng, H. Li, "A high-efficiency discrete logarithm-based multi-proxy blind signature scheme," *International Journal of Network Security*, vol. 20, no. 6, pp. 1200-1205, Nov. 2018.

[21] L. Teng, H. Li, J. Liu, S. Yin, "An efficient and secure Cipher-Text retrieval scheme based on mixed homomorphic encryption and multi-attribute sorting method under cloud environment," *International Journal of Network Security*, vol. 20, no. 5, pp. 872-878, Sep. 2018.

[22] C. C. Wu, S. J. Kao, and M. S. Hwang, "A high quality image sharing with steganography and adaptive authentication scheme," *Journal of Systems and Software*, vol. 84, no. 12, pp. 2196-2207.

[23] N. I. Wu and M. S. Hwang, "Data hiding: Current status and key issues," *International Journal of Network Security*, vol. 4, no. 1, pp. 1-9, Jan. 2007.

[24] S. Yin, J. Liu, L. Teng, "A new krill herd algorithm based on SVM method for road feature extraction," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, no. 4, pp. 997-1005, July 2018.

[25] T. Yuvaraja, R. S. Sabeenian, "Performance analysis of medical image security using steganography based on fuzzy logic," *Cluster Computing*, vol. 5, pp. 1-7, 2018.

Biography

Jing Bi received the M.Eng. degree from Shenyang Institute of Technology, Shenyang, Liaoning province, China in 2004. Her research interests include Image Processing and Recognition, Network Security, Big Data Analysis and Artificial intelligence. Email:910675024@qq.com.

Shoulin Yin received the B.Eng. And M.Eng. degree from Shenyang Normal University, Shenyang, Liaoning province, China in 2013 and 2015 respectively. His research interests include Multimedia Security, Network Security, Filter Algorithm and Data Mining. He received School Class Scholarship in 2015. Yin had published more than 30 international journal papers (SCI or EI journals) on the above research fields. Email:352720214@qq.com.

Hang Li obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hang Li 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. Prof. Li had published more than 30 international journal and international conference papers on the above research fields. Email:lihangsoft@163.com.

Lin Teng received the B.Eng. degree from Shenyang Normal University, Shenyang, Liaoning province, China in 2016. Now, she is a laboratory assistant in Software College, Shenyang Normal University. Her research interests include Multimedia Security, Network Security, Filter Algorithm and Data Mining. Email:ysl352720214@163.com.

Cryptanalysis and Improvement of a Biometric-based Authentication Scheme for Multi-server Architecture

Tao Wan[1], Xiaochang Liu[1], Weichuan Liao[2], and Nan Jiang[1]
(Corresponding author: Tao Wan)
School of Information Engineer, East China Jiaotong University, China[1]
School of Science, East China Jiaotong University, China[2]
(Email: wantao217@163.com)
(Received Nov. 30, 2018; Revised and Accepted Aug. 5, 2019; First Online Sept. 21, 2019)

Abstract

In recent year, with the increasing amount of wireless technologies, biometric-based authentication schemes for multi-server architectures have become more crucial and widely developed. In 2016, Wang *et al.* demonstrated that Mishra *et al.*'s protocol has several drawbacks and proposed an improved authentication scheme of biometric-based architecture using smart card and password. They claimed that their scheme achieves intended security requirements and is more appropriate for practical applications. In this paper, we indicate that their scheme cannot resist session key disclosure, smart card forgery attack, server spoofing attack, user impersonation attack, DoS attack, and no provision of user anonymity. Furthermore, we propose a robust biometric-based authentication scheme using public-key encryption techniques to remove these defects. The performance and functionality comparison shows that our proposed scheme provides the best secure functionality and is computational efficient.

Keywords: Authentication; Biometric; Multi-Server; Security; Smart Card

1 Introduction

With the swift expansion of communication technologies and mobile devices, an increasing number of remote user authentication schemes are usually used to provide services to users. Earlier authentication methods were limited to single-server architecture. However, users need to obtain different services from multiple servers, they not only have to register to different servers, but also need to remember a large number of identities and passwords. Obviously, it is very difficult and unsafe for users to remember and manage multiple information. As a scalable solution, multi-server architecture has been introduced, where the users can register only once at the registration server and avail the services of all associated application servers. Several authors have suggested various authentication protocols for multi-server architecture during the past decade [1, 3, 4, 6, 8, 22, 27] .

Password, smart card and biometrics based authentication verifies the legitimacy of each user and offers the access to network resources. The first remote user password based authentication method was proposed by Lamport [12] . Unfortunately, password based authentication method is vulnerable to some attacks, especially, password guessing attack. Hence, the password with smart card methods have proposed. However, several researches indicated that password with smart card methods are still prone to numerous attacks [9,13,18,21,29] . To solve these problems, many researches have combined the biometric, password and smart card to enhance the security of authentication schemes [14, 17, 19, 23].

In 2009, Wang *et al.* [28] proposed a dynamic ID-based remote user authentication scheme and claimed that their scheme provides user's anonymity. Unfortunately, in 2011, Khan *et al.* [11] presented that Wang's protocol is prone to user anonymity, session key disclosure attack and smart card stolen attack. Furthermore, they proposed an enhanced authentication scheme to overcome the weaknesses of Wang *et al.*'s scheme and is more secure and efficient for practical application environment. In 2012, Chen *et al.* [2] proved that Khan *et al.*'s scheme is still vulnerable to insider attack. To remedy these, they proposed an enhanced authentication scheme and demonstrated their scheme is more secure. In 2013, Jiang *et al.* [10] observed that Chen *et al.*'s scheme achieves neither anonymity nor untraceability, and is sensitive to the identity guessing attack and tracking attack. Then, they proposed an enhanced authentication scheme which achieves user anonymity and untraceablity and claimed that it is a secure and efficient authentication scheme with user privacy preservation which is practical for TMIS. However, Wu and Xu *et al.* [30] proved that Jiang *et al.*'s scheme

still cannot resist off-line password guessing attack, user impersonation attack, denial-of-service attack and so on. They even put forward an improved mutual authentication scheme used for a telecare medical information system. Chuang and Chen *et al.* [5] proposed an efficient and secure dynamic ID-based authentication scheme for TMI systems and demonstrated their scheme overcomes several drawbacks. In 2014, Mishra *et al.* [16] pointed out several drawbacks of Chuang and Chen's protocol, such as, server spoofing attack and Denial-of-Service attack. Furthermore, they proposed an efficient improvement on Chuang and Chen's scheme. In 2016, Wang *et al.* [24] proved that Mishra *et al.*'s protocol was vulnerable to masquerade attack, replay attack and Denial-of-Service attack. They proposed a novel biometric-based multi-server architecture and key-agreement scheme. But, we identify that Wang *et al.*'s scheme is still vulnerable to the server spoofing attack and user impersonation attack. Besides, their scheme cannot resist to session key disclosure, smart card forgery attack, DoS attack and fails to provide user anonymity.

The remainder of this manuscript is organized as follows. We introduce the one-way secure hash function, threat model and biometrics-based fuzzy extractor in Section 2. We review the robust smart card authentication scheme for multi-server architecture proposed by Wang *et al.* in Section 3. We analyze the security flaws of Wang *et al.*'s scheme in Section 4. We present a proposed protocol in Section 5. We compare the performance of our proposed scheme with the previous schemes in Section 6. We conclude this paper in Section 7.

2 Preliminaries

During this section, we briefly describe some concepts relating to secure hash function, threat models and biometrics-based fuzzy extractor as follows.

2.1 One-way Secure Hash Function

A one-way secure hash function $h : \{0,1\}^* \rightarrow \{0,1\}^n$ is considered as cryptographically secure and deterministic algorithm, which takes arbitrary size string x as input and produces a fixed length value $V = h(x) \in \{0,1\}^n$. A secure hash function has the following attributes:

- It is computationally easy to find $V = h(x)$, given $h(\cdot)$ and x.
- It is computationally infeasible to compute x, given V and $h(\cdot)$.
- For given hash code $V = h(x)$ and hash function $h(x)$, it is infeasible to find the input x' such that $h(x') = h(x)$. This property is known as weak collusion resistance property.
- It is difficult to find two inputs $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$. This property is known as strong collusion resistance property.

2.2 Treat Model

For the analysis of security of Wang *et al.*'s scheme and the proposed scheme in this paper, we consider a widely accepted threat model to inspect the security of the proposed protocol that has been considered in most of the existing authentication protocols [7,25]. More details about these threat models are described as below.

- An attacker might be a malicious user or malicious server.
- An attacker can extract the information from the smart card by examining the power consumption or leaked information.
- An attacker is able to eavesdrop all the communications between the parties involved such as a user and a server over a public channel.
- An attacker can trap, insert, modify, resend and delete the eavesdropped transmitted messages.
- An attacker may try to trace the actions of a particular user when any of the transmitted parameter is constant.
- In some situation, an attacker may know the previously established session keys. This presumption help us deal with session key disclosure.

2.3 Biometrics-based Fuzzy Extractor

Here, we briefly discuss the preliminaries about biometrics-based fuzzy extractor used in our scheme. The fuzzy extractor converts the biometric information into two values, which consists of two procedures, namely, *Gen* and *Rep*. More details illustrated as following:

- *Gen* is a generation procedure, which on input biometric data BIO_i, outputs an extracted string P_i and auxiliary string R_i, where $Gen(BIO_i) \rightarrow (R_i, P_i)$.
- *Rep* is a deterministic generation reproduction procedure that allows to recover R_i from the corresponding auxiliary string P_i and any vector BIO_i^* close to BIO_i, where $Rep(BIO_i^*, P_i) \rightarrow R_i$.

  The uniqueness property of a biometric allows its applications in authentication protocols.

3 Review of Wang *et al.*'s Scheme

In this section, we briefly review Wang *et al.*'s biometric-based authentication scheme for multi-server. Three roles participate in this scheme: The user U_i, the server S_j and the registration center RC. There are five phases relating to Wang *et al.*'s scheme, ie. server registration phase, user registration phase, login and authentication phase, password change phase and revocation/re-registration phase. The details are described in the following subsections. Table 1 lists the notations used in this scheme.

Table 1: Notations used in the paper

Symbols	Their meaning
RC	*The registration center*
U_i	*The i_{th} user*
ID_i	*The U_i's identity*
S_j	*The j_{th} application server*
SID_j	*The S_j's identity*
PW_i	*The user U_i's password*
PSK	*Per shared key*
x	*Master secret key*
$h(\cdot)$	*A secure one-way hash function*
$\|$	*Concatenation operation*
$\oplus$	*XOR operation*
SK_{ij}	*Section key shared between U_i and S_j*

3.1 Server Registration Phase

This phase is executed between the application server S_j and the registration center RC. This registration phase consists of the following steps:

Step S1: The server S_j first sends a registration request to the registration center RC.

Step S2: Receiving the registration request from the remote server S_j, the registration center RC assigns the value PSK to the remote server S_j.

3.2 User Registration Phase

When a user wishes to access any services provided by the registered servers, he/she must first register himself/herself. This registration phase consists of the following steps:

Step U1: The user U_i chooses an identity ID_i, password PW_i. Then the user U_i imprints his personal biometric information BIO_i at a sensor. The sensor sketches BIO_i to extract an unpredictable binary string R_i and an auxiliary binary string P_i from $Gen(BIO_i)\rightarrow(R_i, P_i)$. Then, sensor stores P_i in the memory.

Step U2: The user U_i computes $RPW_i = h(PW_i||R_i)$ and sends $\{ID_i, RPW_i\}$ to RC via a secure channel. RC adds a novel entry $< ID_i, N_i = 1 >$ to the database, where N_i means the times of user registration.

Step U3: The registration center RC computes

$$\begin{aligned} A_i &= h(ID_i||x||T_r), \\ B_i &= RPW_i \oplus h(A_i), \\ C_i &= B_i \oplus h(PSK), \\ D_i &= PSK \oplus A_i \oplus h(PSK), \\ V_i &= h(ID_i||RPW_i), \end{aligned}$$

where T_r is the time of user registration time.

Step U4: The registration center RC securely issues the smart card containing $\{B_i, C_i, D_i, V_i\}$ to the user U_i.

Step U5: After receiving the issued smart card, the user U_i stores the P_i into the smart card.

3.3 Login and Authentication Phase

When a legal user U_i wants to access the resources provided by remote server S_j, he/she first attaches the smart card to a device reader, and inputs his/her identity ID_i and password PW_i, and imprints the biometrics BIO_i^* at the sensor. Sensor sketches BIO_i^* and recovers R_i from $Rep(BIO_i^*, P_i)\rightarrow R_i$. Then, as illustrated in Figure 1, the login and authentication mechanism is performed as follows:

Step V1: The user U_i computes $RPW_i = h(PW_i||R_i)$ and checks whether $h(ID_i||RPW_i)$ is equal to V_i. If it holds, the smart card further calculates $h(PSK)=B_i \oplus C_i$, then generates a random nonce N_1 and computes

$$\begin{aligned} AID_i &= ID_i \oplus h(N_1), \\ M_1 &= RPW_i \oplus N_1 \oplus h(PSK), \\ M_2 &= h(AID_i||N_1||RPW_i||SID_j||T_i). \end{aligned}$$

The user U_i sends the login request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$ to the server S_j, where T_i means the timestamp.

Step V2: Upon receiving the message from the user U_i, the server S_j checks whether $T_i - T_j$ is less than $\triangle T$, where T_j is a timestamp. If not, the communication is simply terminated. Otherwise, the server S_j computes

$$\begin{aligned} A_i &= PSK \oplus D_i \oplus h(PSK), \\ RPW_i &= B_i \oplus h(A_i), \\ N_1 &= RPW_i \oplus M_1 \oplus h(PSK). \end{aligned}$$

and verifies whether $h(AID_i||N_1||RPW_i||SID_j||T_i)$ is equal to M_2. If it holds, the server S_j generates a random number N_2, and computes

$$\begin{aligned} SK_{ij} &= h(AID_i|SID_j||N_1||N_2), \\ M_3 &= N_2 \oplus h(AID_i||N_1) \oplus h(PSK), \\ M_4 &= h(SID_j||N_2||AID_i). \end{aligned}$$

Step V3: Furthermore, the server S_j sends the response message $\{SID_j, M_3, M_4\}$ to U_i. Upon getting the response message, the user U_i computes

$$\begin{aligned} N_2 &= M_3 \oplus h(AID_i||N_1) \oplus h(PSK), \\ K_{ij} &= h(AID_i||SID_j||N_1||N_2), \\ N_1 &= B_i \oplus M_1 \oplus h(PSK). \end{aligned}$$

and verifies whether $h(SID_j||N_2||AID_i)$ is equal to M_4. If not, the communication is simply terminated. Otherwise, the user U_i computes M_5 =

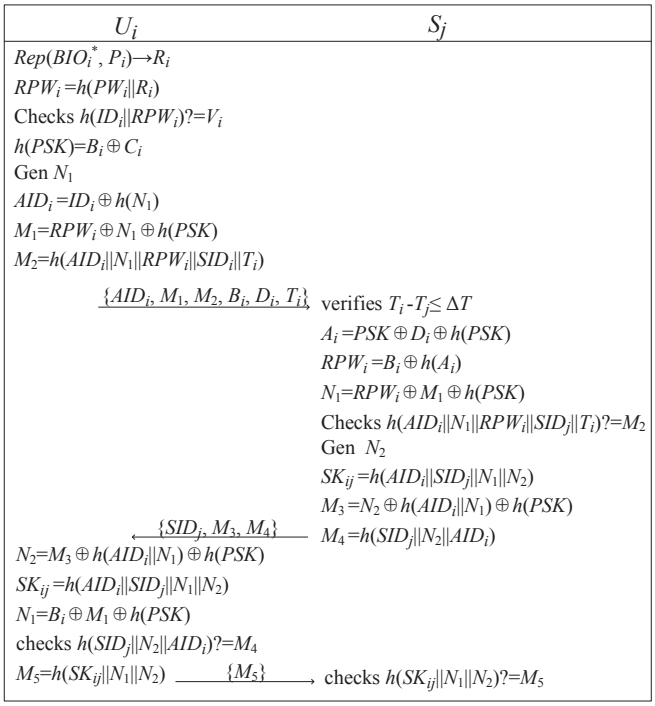


Figure 1: User login and authentication on Wang *et al.*'s Scheme

$h(SK_{ij}||N_1||N_2)$. Then user U_i transmits the message $\{M_5\}$ to the server S_j.

Step V4: Upon getting the message $\{M_5\}$, the server S_j checks whether $h(SK_{ij}||N_1||N_2)$ is similar to M_5. If this condition holds, the server S_j and the user U_i communicates with session key SK_{ij}.

3.4 Password Change Phase

This phase is invoked whenever U_i wants to change his password PW_i to a new password PW_i^{new} .

Step P1: The user U_i inserts his smart card and inputs his identity ID_i and password PW_i, and imprints his biometrics BIO_i^* at sensor. Then the sensor sketches BIO_i^* and recovers R_i from $Rep(BIO_i^*, P_i) \rightarrow R_i$.

Step P2: The smart card calculates $RPW_i = h(PW_i||R_i)$ and checks whether $h(ID_i||RPW_i)$ is similar to V_i. If it holds, smart card asks U_i for a new password.

Step P3: The user U_i input the new password PW_i^{new} and the smart card further computes

$$\begin{aligned} RPW_i^{new} &= h(PW_i^{new}||R_i), \\ B_i^{new} &= B_i \oplus RPW_i \oplus RPW_i^{new}, \\ C_i^{new} &= C_i \oplus RPW_i \oplus RPW_i^{new}, \\ V_i^{new} &= h(ID_i||RPW_i^{new}). \end{aligned}$$

Step P4: The smart card then replaces B_i with B_i^{new}, C_i with C_i^{new}, and V_i with V_i^{new} in the memory.

3.5 User Revocation/Re-registration Phase

If the user U_i wants to revoke his privilege, he needs to send a revocation request message, his smart card and verification message $\{RPW_i\}$ to the registration center RC via a secure channel. The detailed procedure of this phase is shown as follows.

Step R1: RC checks whether U_i is valid. If it holds, RC modifies the corresponding entry by setting $< ID_i, N_i = 0 >$.

Step R2: RC executes the steps described in the section of user registration phase and replaces$< ID_i, N_i = N_i + 1 >$ with $< ID_i, N_i >$ to help U_i re-register.

4 Security Analysis of Wang *et al.*'s Scheme

In Wang *et al.*'s scheme, the security analysis of scheme demonstrated that their scheme satisfies the desirable security requirements. Unfortunately, we find that their scheme still has many vulnerabilities. If an attacker colludes with a registered but malicious server and eavesdrops messages between the user U_i and the server S_j, he can launches session key disclosure, smart card forgery attack, server spoofing attack and user impersonation attack. He also can forge a current timestamp and initiate DoS attack that attempt to make network resource or machines unavailable. Moreover, a user's behavior is tracked because smart card data B_i in the public channel, which can be easily eavesdropped by adversaries. The details are as follows.

4.1 Session Key Disclosure

In Wang *et al.*'s scheme, the registration center RC shares the same pre-shared PSK with all the servers. Once the attacker Z colludes with the registered but malicious server, he can obtain the pre-shared key PSK and launch the session key disclosure. Now we show the reason why Wang *et al.*'s scheme cannot resist to session key disclosure. The attacker intercepts messages $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$, $\{SID_j, M_3, M_4\}$ and calculates the following operations:

$$\begin{aligned} A_i &= PSK \oplus D_i \oplus h(PSK), \\ RPW_i &= B_i \oplus h(A_i), \\ N_1 &= RPW_i \oplus M_1 \oplus h(PSK), \\ N_2 &= M_3 \oplus h(AID_i||N_1) \oplus h(PSK), \\ SK_{ij} &= h(AID_i||SID_j||N_1||N_2), \end{aligned}$$

Now, the attacker Z easily derives the current session key SK_{ij} shared between U_i and S_j. After that, S_k can decrypt all encrypted information between U_i and S_j. Hence, Wang *et al.*'s scheme is vulnerable to session key disclosure.

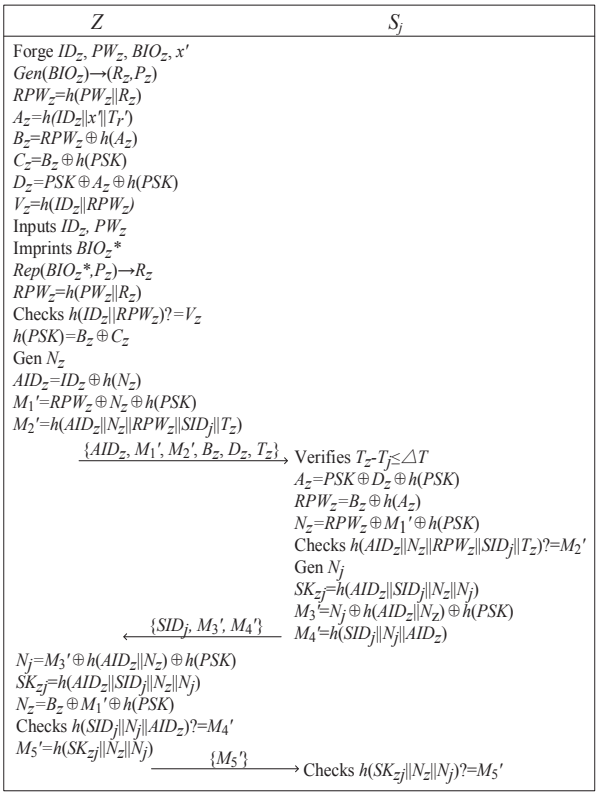


Figure 2: Smart card forgery attack on Wang *et al.*'s Scheme

4.2 Smart Card Forgery Attack

As shown in Wang *et al.*'s scheme, any server has the same pre-shared key PSK. Under the condition that the attacker Z colludes with a malicious S_k, they can forge a smart card to log into any server (*e.g.*, S_j) as shown in Figure 2. The procedure is as following:

- Z forges a new identity ID_z, password PW_z and personal biometric BIO_z, and forges a master key x'. Sensor sketches BIO_z, extracts (R_z, P_z) from $Gen(BIO_z)$, and stores P_z in the memory.

- Z computes

$$\begin{aligned} RPW_z &= h(PW_z||R_z), \\ A_z &= h(ID_z||x'||T_r'), \\ B_z &= RPW_z \oplus h(A_z), \\ C_z &= B_z \oplus h(PSK), \\ D_z &= PSK \oplus A_z \oplus h(PSK), \\ V_z &= h(ID_z||RPW_z), \end{aligned}$$

  then the forged smart card containing $\{B_z, C_z, D_z, V_z, P_z\}$.

- Z inserts the forged smart card and input identity ID_z, password PW_z and personal biometric BIO_z, sensor sketches BIO_z recovers R_z from $Rep(BIO_z, P_z) \rightarrow R_z$.

- Z computes $RPW_z = h(PW_z||R_z)$ and checks whether $h(ID_z||RPW_z)$ is equal to V_z. Obviously, $h(ID_z||RPW_z)$ is equal to V_z. Then, Z computes $h(PSK) = B_z \oplus C_z$, generates a random number N_z, computes

$$\begin{aligned} AID_z &= ID_z \oplus h(N_z), \\ M_1' &= RPW_z \oplus N_z \oplus h(PSK), \\ M_2' &= h(AID_z||N_z||RPW_z||SID_j||T_z). \end{aligned}$$

  Then, the forged smart card send the request message $\{AID_z, M_1', M_2', B_z, D_z, T_z\}$ to S_j via a public channel.

Upon receiving the message $\{AID_z, M_1', M_2', B_z, D_z, T_z\}$, S_j verifies whether $T_z - T_j$ is less than $\triangle T$. If the condition holds, the server S_j computes

$$\begin{aligned} A_z &= PSK \oplus D_z \oplus h(PSK), \\ RPW_z &= B_z \oplus h(A_z), \\ N_z &= RPW_z \oplus M_1' \oplus h(PSK), \end{aligned}$$

and checks whether $h(AID_z||N_z||RPW_z||SID_j||T_z)$ is equal to M_2'. The server S_j generates a random number N_j , and computes

$$\begin{aligned} SK_{zj} &= h(AID_z||SID_j||N_z||N_j), \\ M_3' &= N_j \oplus h(AID_z||N_z) \oplus h(PSK), \\ M_4' &= h(SID_j||N_j||AID_z). \end{aligned}$$

Finally, S_j sends the message $\{SID_j, M_3', M_4'\}$ to the attacker Z.

When receiving the replay message $\{SID_j, M_3', M_4'\}$, Z computes

$$\begin{aligned} N_j &= M_3' \oplus h(AID_z||N_z) \oplus h(PSK), \\ SK_{zj} &= h(AID_z||SID_j||N_z||N_j) \end{aligned}$$

and $N_z = B_z \oplus M_1' \oplus h(PSK)$. Obviously, $h(SID_j||N_j||AID_z)$ is equal to M_4'. The attacker Z computes $M_5' = h(SK_{zj}||N_z||N_j)$ and sends $\{M_5'\}$ to the server S_j.

At last, the attacker successfully logs into the server S_j using the forged smart card. Therefore, Wang *et al.*'s scheme cannot resist smart card forgery attack.

4.3 Server Spoofing Attack

In the server registration phase, RC transmits the same pre-shared key PSK to every server, thus an authorized but malicious server S_k can impersonate as any server (*e.g.*, S_j) to deceive any legal user after he intercepts the request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$. S_k masquerades as the server S_j to spoof U_i in the following way.

- S_k can retrieve

$$\begin{aligned} A_i &= PSK \oplus D_i \oplus h(PSK), \\ RPW_i &= B_i \oplus h(A_i), \\ N_1 &= RPW_i \oplus M_1 \oplus h(PSK), \end{aligned}$$

by capturing the message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$.

- S_k generates N_2', calculates

$$\begin{aligned} SK_{ij}' &= h(AID_i||SID_j||N_1||N_2'), \\ M_3' &= N_2' \oplus h(AID_i||N_1) \oplus h(PSK), \\ M_4' &= h(SID_j||N_2'||AID_i), \end{aligned}$$

then sends $\{SID_j, M_3', M_4'\}$ to U_i via a public channel.

- After receiving the message, U_i computes

$$\begin{aligned} N_2' &= h(AID_i||N_1) \oplus M_3' \oplus h(PSK), \\ SK_{ij}' &= h(AID_i||SID_j||N_1||N_2'). \end{aligned}$$

Then the user U_i verifies the condition

$$h(SID_j||N_2'||AID_i)? = M_4'.$$

Evidently, this condition holds. The user U_i mistakenly thinks that he is communicating with S_j.

At last, the authorized malicious server S_k can successfully launch the server spoofing attack.

4.4 User Impersonation Attack

As shown in Wang *et al.*'s scheme, the user U_i transmits the request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$ to the server S_j, S_j can retrieve the user's identity $ID_i = AID_i \oplus h(N_1)$ through computing

$$\begin{aligned} A_i &= PSK \oplus D_i \oplus h(PSK), \\ RPW_i &= B_i \oplus h(A_i), \\ N_1 &= RPW_i \oplus M_1 \oplus h(PSK). \end{aligned}$$

Once the server reveals ID_i and RPW_i to the attacker Z, Z can impersonate as the user, the details are shown as below.

- The attacker Z generates a random number N_1' and computes

$$\begin{aligned} AID_i' &= ID_i \oplus h(N_1'), \\ M_1' &= RPW_i \oplus N_1' \oplus h(PSK), \\ M_2' &= h(AID_i'||N_1'||RPW_i||SID_j||T_i'). \end{aligned}$$

Finally, Z delivers his login request message $\{AID_i', M_1', M_2', B_i, D_i, T_i'\}$ to the server S_j.

- Upon the server S_j receiving the message, S_j checks whether T_j-T_i'<=$\triangle T$ is valid. If the condition holds, S_j computes

$$\begin{aligned} A_i &= PSK \oplus D_i \oplus h(PSK), \\ RPW_i &= B_i \oplus h(A_i), \\ N_1' &= RPW_i \oplus M_1' \oplus h(PSK). \end{aligned}$$

S_j checks whether $h(AID_i'||N_1'||RPW_i||SID_j||T_i)$ is similar to M_2'.

- The server S_j generates a random number N_2, computes

$$\begin{aligned} SK_{ij}' &= h(AID_i'||SID_j||N_1'||N_2), \\ M_3' &= N_2 \oplus h(AID_i'||N_1') \oplus h(PSK), \\ M_4' &= h(SID_j||N_2||AID_i'), \end{aligned}$$

and sends $\{SID_j, M_3', M_4'\}$ to Z over a public channel.

- The attacker Z computes

$$\begin{aligned} N_2 &= M_3' \oplus h(AID_i'||N_1') \oplus h(PSK), \\ SK_{ij}' &= h(AID_i'||SID_j||N_1'||N_2), \\ N_1' &= B_i \oplus M_1' \oplus h(PSK). \end{aligned}$$

Obviously, $h(SID_j||N_2\ ||AID_i')$ is equal to M_4'. Then, the attacker Z calculates $M_5' = h(SK_{ij}'||N_1'||N_2)$ and sends $\{M_5'\}$ to S_j via a public channel.

- The server S_j checks whether $h(SK_{ij}'||N_1'||N_2)$ is equal to M_5'. If it holds, S_j uses the session key SK_{ij}' to communicate with Z and believes that he is the legal user U_i.

Thus, Wang *et al.*'s scheme cannot resist to user impersonation attack.

4.5 Denial of Service Attack

From the login and authentication phase of Wang *et al.*'s scheme, we find that any attacker Z who colludes with the malicious server can easily forge a login request message and replay it to the server S_j. In Wang *et al.*'s scheme, the attacker can launch DoS attack as described below:

- Upon intercepting the message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$, the attacker Z computes

$$\begin{aligned} A_i &= PSK \oplus D_i \oplus h(PSK), \\ RPW_i &= B_i \oplus h(A_i), \\ N_1 &= RPW_i \oplus M_1 \oplus h(PSK). \end{aligned}$$

- The attacker Z generates a current timestamp T_i' and calculates $M_2' = h(AID_i||N_1||RPW_i||SID_j||T_i')$. Z sends $\{AID_i, M_1, M_2', B_i, D_i, T_i'\}$ to S_j.

- Upon receiving the message from Z, S_j computes

$$\begin{aligned} A_i &= PSK \oplus D_i \oplus h(PSK), \\ RPW_i &= B_i \oplus h(A_i), \\ N_1 &= RPW_i \oplus M_1 \oplus h(PSK) \end{aligned}$$

and verifies whether

$$h(AID_i||N_1||RPW_i||SID_j||T_i')$$

is similar to M_2'. Obviously, the verification holds.

- S_j generates a number N_j and computes

$$\begin{aligned} SK_{ij} &= h(AID_i||SID_j||N_1||N_j), \\ M_3 &= N_2 \oplus h(AID_i||N_1) \oplus h(PSK), \\ M_4 &= h(SID_j||N_2||AID_i). \end{aligned}$$

- S_j sends message $\{SID_j, M_3, M_4\}$ to the user U_i. The attacker Z will intercept the message to terminate the communication.

By this way, the attacker can launch DoS attack on the server S_j, which will result in the computing and communication loss of the server.

4.6 No Provision of User Anonymity

The user anonymity is a desirable property for remote user authentication. Generally, the scheme with user anonymity contains two aspects of content, one is the user's real identity cannot be revealed by the attacker, another is that the user cannot be traced by the attacker. In Wang *et al.*'s scheme, Any server authenticated with the user can recover the identity of the user.Any server authenticated with the user can recover the identity of the user through computing

$$\begin{aligned} A_i &= PSK \oplus D_i \oplus h(PSK), \\ RPW_i &= B_i \oplus h(A_i), \\ N_1 &= RPW_i \oplus M_1 \oplus h(PSK), \\ ID_i &= AID_i \oplus h(N_1), \end{aligned}$$

which D_i and AID_i are intercepted from the message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$. Thus, the identity of the user is leaked to the server. Moreover, in each login phase, the user U_i submits the login request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$ to the server S_j. On this message, $B_i = RPW_i \oplus h(A_i)$ and $D_i = PSK \oplus A_i \oplus h(PSK)$ are unique for each user. The attacker can distinguish whether two sessions are launched by the same user. Therefore, the attacker can trace the user by B_i and D_i. Accordingly, Wang *et al.'s* scheme fails to preserve user anonymity.

5 The Proposed Protocol

In this section, based on the cryptanalysis of Wang *et al.*'s scheme, we present our robust biometrics-based multi-server authentication scheme with smart card using public-key encryption technique, where Pub_{sj} is the public key of S_j, Pri_{sj} is the secret key of S_j. The proposed scheme consists of three phases: Registration phase, login and authentication phase and password change phase . There are also three participants: The user U_i, the server S_j and the registration center RC.

5.1 Registration Phase

In our proposed protocol, the registration phase consists of two sub-phases, the server registration phase and the user registration phase. In this phase, the server and the user should register themselves to the registration center RC and obtains secret information to initial system.

5.1.1 Server Registration Phase

The server S_j sends a registration request to RC in order to become an authorized server. This registration process consists of following steps:

Step S1: The server S_j sends a registration request message $\{SID_j\}$ to RC.

Step S2: The registration center RC replies with $\{h(PSK||SID_j)\}$ to the server S_j, which can be used in further phases of authentication.

5.1.2 User Registration Phase

When a user wants to access the services of servers, he must register himself, as shown in Figure 3. This registration process according to the following steps:

Step R1: The user U_i freely selects his identity ID_i, which uniquely identities the user's identity, password PW_i and scans his biometrics BIO_i at sensor terminal to gets R_i from $Gen(BIO_i) \rightarrow (R_i, P_i)$. Then the user U_i generates a random number b_i and computes $AID_i = h(ID_i||b_i)$ and $RPW_i = h(PW_i||R_i||b_i)$. At last, the user U_i sends a request message $\{AID_i, RPW_i\}$ to S_j via a secure channel.

Step R2: Upon getting the message, RC computes

$$\begin{aligned} B_{ij} &= h(AID_i||h(PSK||SID_j)), \\ C_{ij} &= B_{ij} \oplus RPW_i, \\ V_i &= h(AID_i||RPW_i). \end{aligned}$$

Step R3: The RC selects a base point G and stores $\{< SID_j, C_{ij} >, V_i, G, h(\cdot)\}$ into the smart card and delivers it to the user U_i via a secure channel.

Step R4: Upon getting the message, the user U_i stores $\{b_i, P_i\}$ into the smart card.

5.2 Login and Authentication Phase

When a user U_i wants to access the services of remote server S_j, he launches the login request by inserting smart card , and inputting ID_i and PW_i. Next, the user U_i imprints his biometric information BIO_i at a sensor. After that, sensor sketches user U_i's biometric information BIO_i and recovers the unpredictable binary string R_i from $Rep(BIO_i, P_i) \rightarrow R_i$. Then, as shown in Fig 4, the login and authentication procedure is performed as follows:

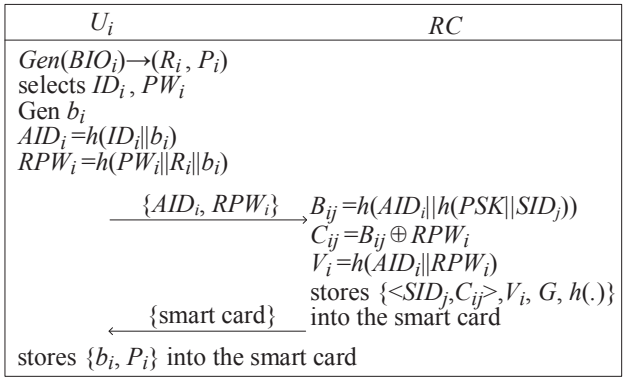


Figure 3: User registration phase of our scheme

Step L1: The smart card computes $AID_i = h(ID_i||b_i)$, $RPW_i = h(PW_i||R_i||b_i)$, and verifies whether V_i is equal to $h(AID_i||RPW_i)$. If V_i is invalid, smart card terminates the communication; otherwise, the user U_i generates a random number N_1 and calculates

$$\begin{aligned} B_{ij} &= RPW_i \oplus C_{ij}, \\ D_i &= N_1 \cdot G, \\ F_{ij} &= B_{ij} \oplus D_i, \\ M_1 &= E_{Pub_{sj}}(AID_i||T_i), \\ M_2 &= h(AID_i||B_{ij}||D_i||T_i). \end{aligned}$$

Then the user U_i sends the login request message $\{F_{ij}, M_1, M_2, T_i\}$ to the server S_j, where T_i is a current timestamp.

Step L2: Upon receiving the message from the user U_i, the server S_j checks whether T_i - T_j is less than $\triangle T$, where $\triangle T$ is the time interval and T_j is the time when S_j receives the login request message. The server S_j computes

$$\begin{aligned} AID_i||T_i &= D_{Pri_{sj}}(M_1), \\ B_{ij} &= h(AID_i||h(PSK||SID_j)), \\ D_i &= B_{ij} \oplus F_{ij}, \end{aligned}$$

and verifies whether the condition M_2 is equal to $h(AID_i||B_{ij}||D_i||T_i)$. If the condition holds, the server S_j authenticates the user U_i, otherwise the process can be terminated.

Step L3: The server S_j further generates a random number N_2 and computes

$$\begin{aligned} D_j &= N_2 \cdot G, \\ P_j &= N_2 \cdot D_i, \\ SK_{ij} &= h(AID_i||SID_j||P_j||D_j), \\ M_3 &= h(SK_{ij}||AID_i||D_j). \end{aligned}$$

Furthermore, the server S_j sends the response message $\{M_3, D_j\}$ to the user U_i.

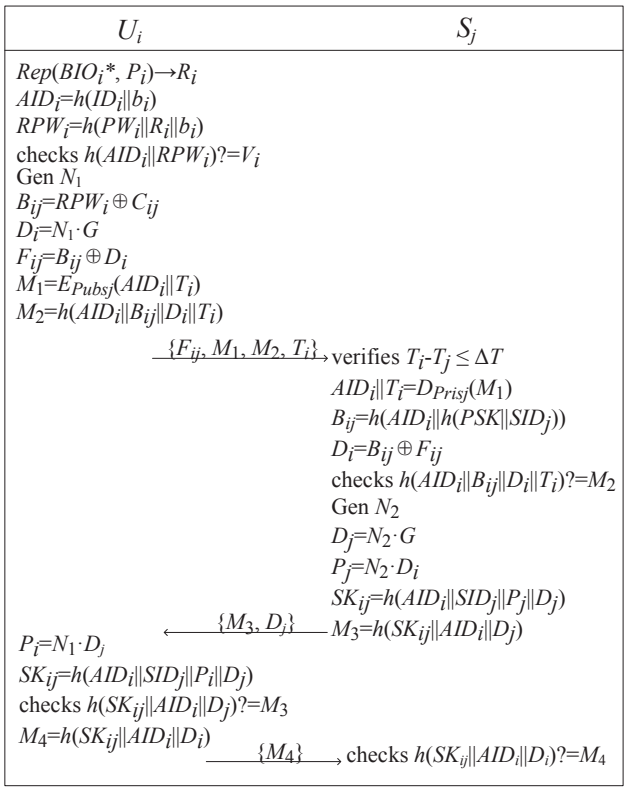


Figure 4: Login and authentication phase of our scheme

Step L4: After receiving the message $\{M_3, D_j\}$, the user U_i computes

$$\begin{aligned} P_i &= N_1 \cdot D_j, \\ SK_{ij} &= h(AID_i||SID_j||P_i||D_j), \end{aligned}$$

and verifies whether the condition M_3 is similar to $h(SK_{ij}||AID_i||D_j)$. If the condition holds, the user U_i authenticates the remote server S_j, otherwise the process is terminated. Then, the user computes $M_4 = h(SK_{ij}||AID_i||D_i)$ and sends the message $\{M_4\}$ to the server S_j.

Step L5: Upon receiving the message $\{M_4\}$, the server S_j verifies whether M_4 is equal to $h(SK_{ij}||AID_i||D_i)$. If not , the server S_j terminates the communication. Otherwise, the user U_i and the server S_j can use the current session key SK_{ij} for securing communication.

5.3 Password Change Phase

This procedure invokes when a user U_i wishes to update his password. The user U_i can change his password as follows:

Step P1: The user U_i inputs ID_i and PW_i, and imprints his biometrics BIO_i. The sensor sketches BIO_i and recovers R_i from $Rep(BIO_i, P_i) \rightarrow R_i$.

Step P2: The smart card computes

$$\begin{aligned} AID_i &= h(ID_i||b_i), \\ RPW_i &= h(PW_i||R_i||b_i), \end{aligned}$$

and then verifies whether V_i is similar to $h(AID_i||RPW_i)$. If this verification is valid, the smart card asks user U_i for a new password. Otherwise, password change phase is terminated immediately by the smart card.

Step P3: The user U_i chooses a new password PW_i^{new} and generates a random number b_i^{new}. Then U_i computes

$$\begin{aligned} AID_i^{new} &= h(ID_i||b_i^{new}), \\ RPW_i^{new} &= h(PW_i^{new}||R_i||b_i^{new}), \\ C_i^{new} &= B_{ij} \oplus RPW_i^{new}, \\ V_i^{new} &= h(AID_i^{new}||RPW_i^{new}). \end{aligned}$$

Step P4: In the memory, smart card respectively replaces C_i with C_i^{new} and V_i with V_i^{new}.

6 Analysis of the Proposed Protocol

In this section, we first present security analysis of our scheme, and then analyze its performance efficiency by comparing it with previous related works.

6.1 User Anonymity

In our scheme, the real identity of user is not revealed throughout all the phases of communication. In the user registration phase, U_i submits $AID_i = h(ID_i||b_i)$ to RC, which the real identity is protected with a one-way hash function and random number b_i. During the login phase, the messages $\{F_{ij}, M_2, T_i\}$, $\{M_3, D_j\}$ and $\{M_4\}$ are converted as dynamic in the form of $D_i = N_1 \cdot G$ and $D_j = N_2 \cdot G$, where N_1 and N_2 are random numbers. The message $\{M_1\}$ is converted as dynamic by freshness timestamp T_i. All the messages between the user and the server are dynamic and dose not disclose the identity of U_i. Hence, our scheme can provide user anonymity.

6.2 Resistance to User Impersonation Attack

Consider a scenario where the attacker U_z acts as a legitimate one and proceeds with the authentication procedures. If the attacker U_z wants to impersonate a legitimate user U_i, he requires to build a login request message $\{F_{ij}, M_1, M_2, T_i\}$, where $F_{ij} = B_{ij} \oplus D_i$, $M_1 = E_{Pub_{sj}}(AID_i||T_i)$ and $M_2 = h(AID_i||B_{ij}||D_i||T_i)$. However, the attacker cannot compute $D_i = N_1 \cdot G$ because N_1 is the user generated random number. Moreover, in order to compute AID_i and B_{ij}, the attacker requires user's identity ID_i and password PW_i, which are unobtainable. So our scheme is secure against the user impersonation attack.

6.3 Resistance to Server Spoofing Attack

In the proposed scheme, if the malicious server S_k wants to authenticate with the user U_i by impersonating as the server S_j, S_k needs to compute $B_{ij} = h(AID_i||h(PSK||SID_j))$. Although S_k can capture parameters AID_i and SID_j, it is impossible for S_k to retrieve the pre-share key PSK from the registration center RC. Because on the server registration phase, the registration center RC transmits $h(PSK||SID_j)$ to S_k, rather than PSK. Therefore, our proposed scheme withstands the server spoofing attack.

6.4 Resistance to Session Key Disclosure

In our scheme, the session key is defined as $SK_{ij} = h(AID_i||SID_j||P_j||D_j) = h(AID_i||SID_j||P_i||D_j)$, where $P_j = P_i = N_1 \cdot N_2 \cdot G$ and $D_j = N_2 \cdot G$ with randomly chosen number N_1 and N_2 . We can see N_1 and N_2 are random nonce generated by user and server. Obviously, attacker cannot get N_1 and N_2. Moreover, the attacker cannot get AID_i due to only the server S_j can decrypt the message $M_1 = E_{Pub_{sj}}(AID_i||T_i)$ using the private key Pri_{sj} of the server. Thus, our scheme can resist session key disclosure.

6.5 Resistance to Smart Card Forgery Attack

In our proposed scheme, the smart card contains $\{C_{ij}, V_i, b_i, P_i\}$. If the attacker attempts to forge smart card, he forges a new identity ID_z, password PW_z and personal biometric BIO_z. Sensor sketches BIO_z, extracts (R_z, P_z) from $Gen(BIO_z) \rightarrow (R_z, P_z)$, and stores P_z into smart card. The attacker generates a random number b_z, and calculates $AID_z = h(ID_z||R_z||b_z)$ and $RPW_z = h(PW_z||R_z||b_z)$. To forge parameter C_{zj}, the attacker attempt to compute $B_{zj} = h(AID_z||h(PSK||SID_j))$. Unfortunately, the attacker cannot retrieve PSK since RC calculates $h(PSK||SID_j)$ for each S_j. So, the attacker cannot forge C_{zj}. Thus, our scheme can resist smart card forgery attack.

6.6 Resistance to Privileged Insider Attack

During user registration phase of our proposed scheme, U_i dose not submits identity ID_i and password PW_i in plaintext form to the registration server RC. U_i submits $AID_i = h(ID_i||b_i)$ and $RPW_i = h(PW_i||R_i||b_i)$ to RC, where b_i is a random number generated by the user U_i. Hence, an insider cannot obtain the original credentials of any user. In this way, our proposed protocol attains resistance to privileged insider attacks.

Table 2: Efficiency Comparison

	User side	Server side	Total	Times(ms)
Reddy et al.[20]	$8T_h$+$2T_{epm}$	$5T_h$+$1T_{epm}$	$13T_h$+$3T_{epm}$	6.693
Lu et al.[14]	$4T_h$+$3T_{re}$	$14T_h$+$3T_{rd}$	$18T_h$+$3T_{re}$+$3T_{rd}$	12.1689
Mishra et al.[15]	$6T_h$+$2T_{epm}$	$10T_h$+$1T_{epm}$	$16T_h$+$3T_{epm}$	6.7148
Wang et al.[24]	$12T_h$	$8T_h$	$20T_h$	0.08
Our scheme	$9T_h$+$1T_{re}$	$5T_h$+$1T_{rd}$	$14T_h$+$1T_{re}$+$1T_{rd}$	4.0533

6.7 Resistance to Replay Attack

If the attacker intercepts the communication message $\{F_{ij},\ M_1,\ M_2,\ T_i\}$ between U_i and S_j, he/she transmits $\{F_{ij},\ M_1,\ M_2,\ T_i'\}$ to the server S_j, where T_i' is a current timestamp. Upon receiving the response message, S_j computes $M_2' = h(AID_i||B_{ij}||D_i||T_i')$ and verifies whether M_2' is equal to M_2. Here, S_j identifies it as a fake response from the malicious user due to $M_2' \neq M_2$ and terminates the session immediately. Hence, our protocol is secure against replay attack.

6.8 Resistance to Password Guessing Attack

The attacker may try to guess the password PW_i from the extracted smart card stored parameters $\{C_{ij},\ V_i,\ h(\cdot)\}$. The stored parameter contains the password PW_i in the form $RPW_i = h(PW_i||R_i||b_i)$, where R_i froms $Gen(BIO_i) \rightarrow (R_i, P_i)$. The attacker attempts to verify the condition $V_i? = h(AID_i||RPW_i)$ while constantly guessing PW_i. The attacker needs the value of ID_i and R_i of U_i in order to achieve the password guessing attack. However, the value of R_i is nowhere stored and the attacker cannot know ID_i. As a result, he cannot guess PW_i. Therefore, our scheme resist to password guessing attack.

6.9 Perfect Forward Secrecy

The session key of the proposed protocol is computed as $S_{ij} = h(AID_i||SID_j||P_j||D_j) = h(AID_i||SID_j||P_i||D_j)$, where $P_j = P_i = N_1 \cdot N_2 \cdot G$ and $D_j = N_2 \cdot G$. Although the long term key is compromised with the attacker, he still cannot construct a valid session key due to following reason. The parameter P_i, P_j and D_j are dynamic due to its association with random generated number N_1 and N_2, which is not possible to extract. Therefore, the proposed protocol provides perfect forward secrecy.

6.10 Performance and Functionality Comparisons

In this section, we compare our proposed protocol with several related schemes [14, 15, 20, 24]. In Table 2, we provide the comparison based on the key security of these schemes, while we compare their efficiency in terms of computation. According to Kilinc *et al.*'s [31] estimation, the average running time of T_h is about 0.0004ms, T_{re} is 3.8500, T_{rd} is 0.1925ms and T_{epm} is 2.229ms. Table 2 illustrates the comparative performance of our improved scheme and previously proposed schemes. From that, we can see our proposed scheme is more efficient than Reddy *et al.*'s scheme, Lu *et al.*'s scheme and Mishra *et al.*'s scheme. The following notations are used in Table 2.

- T_h: The execution time of one-way hash;
- T_{re}: RSA encryption;
- T_{rd}: RSA decryption;
- T_{epm}: The time for executing a scalar multiplication operation of elliptic curve.

We perform a comparative functional analysis of previous schemes, which is illustrated in Table 3. For fair comparison, we use the objective third-party evaluation metrics, where refer to Wang *et al.*'s scheme [26]. As illustrated in Table 3, our scheme provides all the 15 criteria while maintaining reasonable efficiency, all the other schemes fail to achieve at least one critical criterion. Thus, we can find that our proposed scheme is more secure and provides more functionality requirements than the other related schemes.

7 Conclusions

In this paper, we analyzed Wang *et al.*'s smart card based multi-server authentication scheme. Our analysis reveals its inherent security vulnerabilities, i.e., session key disclosure, smart card forgery attack, server spoofing attack, user impersonation attack, DoS attack and no provision of user anonymity. In addition, this paper proposed a robust biometrics-based multi-server authentication scheme with smart cards using public-key encryption techniques. The mutual authentication of the proposed protocol achieved significant features such as biometric authentication, public-key encryption techniques, with less computational and communication cost. Furthermore, the comparison results evidently indicate that our protocol is more secure than other schemes. Thus, our protocol is more feasible for practical applications.

Table 3: Security Comparison

	Reddy et al.[20]	Lu et al.[14]	Mishra et al.[15]	Wang et al.[24]	Our scheme
C1: *No password verifier table*	Yes	Yes	Yes	Yes	Yes
C2: *Password Friendly*	Yes	Yes	Yes	Yes	Yes
C3: *No password exposure*	Yes	Yes	Yes	Yes	Yes
C4: *No smart card loss attack*	Yes	Yes	Yes	No	Yes
C5: *Resistance to known attack*	No	No	No	No	Yes
C6: *Sound repairability*	Yes	Yes	Yes	Yes	Yes
C7: *Provide key agreement*	No	Yes	Yes	No	Yes
C8: *No clock synchronization*	Yes	Yes	Yes	No	Yes
C9: *Timely typo detection*	Yes	Yes	Yes	Yes	Yes
C10: *Mutual authentication*	Yes	No	No	No	Yes
C11: *User anonymity*	Yes	No	Yes	No	Yes
C12: *Forward secercy*	Yes	Yes	No	No	Yes
C13: *Resistance to insider attack*	No	Yes	Yes	No	Yes
C14: *Resistance to verifier attack*	Yes	Yes	No	Yes	Yes
C15: *Provide re-registration phase*	No	No	No	Yes	Yes

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 61962022), Key Research and Development Plan of Jiangxi Province (No. 20192BBE50077), and Excellent Scientific and Technological Innovation Teams of Jiangxi Province (No. 20181BCB24009).

References

[1] P. Chandrakar and H. Om, "A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC," *Computer Communications*, vol. 110, 2017.

[2] H. M. Chen, J. W. Lo, C. K. Yeh, "An efficient and secure dynamic ID-based authentication scheme for telecare medical information systems," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3907–3915, 2012.

[3] T. Y. Chen, M. S. Hwang, C. C. Lee, J. K. Jan, "Cryptanalysis of a secure dynamic ID based remote user authentication scheme for multi-server environment," in *Fourth International Conference on Innovative Computing, Information and Control (ICICIC'09)*, pp. 725–728, IEEE, 2009.

[4] 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," *Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, Nov. 2013.

[5] M. C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *International Journal of Network Security*, vol. 18, no. 5, pp. 997–1000, 2014.

[6] 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.

[7] H. Guo, P. Wang, X. Zhang, Y. Huang, and F. Ma, "A robust anonymous biometric-based authenticated key agreement scheme for multi-server environments," *Plos One*, vol. 12, no. 11, pp. e0187403, 2017.

[8] D. He and S. Wu, "Security flaws in a smart card based authentication scheme for multi-server environment," *Wireless Personal Communications*, vol. 70, no. 1, pp. 323–329, 2013.

[9] M. S. Hwang, Li-Hua Li, "A New Remote User Authentication Scheme Using Smart Cards", *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, Feb. 2000.

[10] Q. Jiang, Z. Ma, and G. Li, "A privacy enhanced authentication scheme for telecare medical information systems," *Journal of Medical Systems*, vol. 37, no. 1, pp. 9897, 2013.

[11] M. K. Khan, S. K. Kim, and 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] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 2, no. 24, pp. 770–772, 1981.

[13] C. H. Ling and M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN," *International Journal of Network Security*, vol. 19, no. 2, pp. 177–181, 2017.

[14] Y. Lu, L. Li, H. Peng, and Y. Yang, "A biometrics and smart cards-based authentication scheme for multi-server environments," *Security & Communication Networks*, vol. 8, no. 17, pp. 3219–3228, 2015.

[15] D. Mishra, "Design and analysis of a provably secure multi-server authentication scheme," *Wireless Personal Communications*, vol. 86, no. 3, pp. 1095–1119, 2016.

[16] 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, no. 18, pp. 8129–8143, 2014.

[17] J. Moon, D. Lee, Y. Lee, and D. Won, "Improving biometric-based authentication schemes with smart card revocation/reissue for wireless sensor networks," *Sensors*, vol. 17, no. 5, pp. 1–24, 2017.

[18] S. Qiu, G. Xu, H. Ahmad, and Y. Guo, "An enhanced password authentication scheme for session initiation protocol with perfect forward secrecy," *Plos One*, vol. 13, no. 3, pp. e0194072, 2018.

[19] C. Quan, J. Jung, J. Kim, Q. Sun, D. Lee, and D. Won, "Cryptanalysis and improvement of a biometric and smart card based remote user authentication scheme," in *International Conference on Ubiquitous Information Management and Communication*, pp. 50, 2017.

[20] A. G. Reddy, A. K. Das, V. Odelu, and K. Y. Yoo, "An enhanced biometric based authentication with key-agreement protocol for multi-server architecture based on elliptic curve cryptography," *Plos One*, vol. 11, no. 5, pp. e0154308, 2016.

[21] J. Srinivas, S. Mukhopadhyay, and D. Mishra, "A self-verifiable password based authentication scheme for multi-server architecture using smart card," *Wireless Personal Communications*, vol. 96, no. 18, pp. 1–25, 2017.

[22] W. Tao, J. Nan, and M. A. Jianfeng, "Cryptanalysis of two dynamic identity based authentication schemes for multi-server architecture," *China Communications*, vol. 11, no. 11, pp. 125–134, 2014.

[23] W. Tao, J. Nan, and M. A. Jianfeng, "Cryptanalysis of a biometric-based multi-server authentication scheme," *International Journal of Security and its Application*, vol. 10, no. 2, pp. 163–170, 2016.

[24] C. Wang, X. Zhang, and Z. Zheng, "Cryptanalysis and improvement of a biometric-based multi-server authentication and key agreement scheme," *Plos One*, vol. 11, no. 2, pp. e0149173, 2016.

[25] D. Wang, D. He, W. Ping, and C. H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Transactions on Dependable Secure Computing*, vol. 12, no. 4, pp. 428–442, 2015.

[26] D. Wang, W. Li, and W. Ping, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 1–1, 2018.

[27] D. Wang, L. I. W. Ting, P. Wang, "Crytanalysis of three anonymous authentication schemes for multi-server environment," *Journal of Software*, vol. 29, no. 7, pp. 1937–1952, 2018.

[28] Y. Y. Wang, J. Y. Liu, F. X. Xiao, and J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 32, no. 4, pp. 583–585, 2009.

[29] H. Wijayanto and M. S. Hwang, "Improvement on timestamp-based user authentication scheme with smart card lost attack resistance," *International Journal of Network Security*, vol. 17, no. 2, pp. 160–164, 2015.

[30] F. Wu and L. Xu, "Security analysis and improvement of a privacy authentication scheme for telecare medical information systems," *Journal of Medical Systems*, vol. 37, no. 4, pp. 9958, 2013.

[31] T. Yanik and H. H. Kilinc, "A survey of sip authentication and key agreement schemes," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 1005–1023, 2014.

Biography

Tao Wan received her B.S. degree in Mathematics from Hunan University, Changsha, China, and received her M.S. and Ph.D. degree in Computer Science from Xidian University, Xi'an, China. She is now an associate professor at East China Jiaotong University. Her research interests include cryptography, network and information security, e-commerce security technology.

Xiaochang Liu received her B.S. degree in Software Engineering from North University of China, Taiyuan, China. She is currently a M.S. candidate at East China Jiaotong University, Nanchang, China. Her research interests include network and information security, e-commerce security technology.

Weichuan Liao received his B.S. and M.S. degree in Mathematics from Hunan University, Changsha, China. He is now an associate professor at East China Jiaotong University. His research interests include cryptography, network and information security.

Nan Jiang received his Ph.D. degree in Computer Application Technology from Nanjing University of Aeronautics and Astronautics, Nanjing, China. Now he is an associate professor at East China Jiaotong University. From 2013 to 2014 he is a research scholar in Complex Networks and Security Research Lab at Virginia Tech. His research interests include wireless sensor networks, wireless protocol and architecture, distributed computing and complex network theory.

Efficient Group Signature Scheme without Pairings

Ke Gu[1,2], Dianxing Liu[2], and Bo Yin[1]
(Corresponding author: Ke Gu)
School of Computer and Communication Engineering, Changsha University of Science and Technology[1]
Wangjiali Rd, Tianxin district, Changsha, Hunan Province 410114, China
Key Laboratory of Network Crime Investigation of Hunan Provincial Colleges, Hunan Police Academy[2]
(Email: gk4572@163.com)
(Received Dec. 7, 2018; Revised and Accepted June 18, 2019; First Online July 31, 2019)

Abstract

Although currently many group signature schemes have been proposed, most of them are constructed on pairings. In this paper, we present an efficient group signature scheme without pairings under the model of verifier-local revocation, which is based on the modified EDL signature (first proposed by D. Chaum *et al.* in Crypto 92). Compared with other group signature schemes, the proposed scheme does not employ pairing computation and has the constant signing time and signature size, whose security can be reduced to the computational Diffie-Hellman (CDH) assumption in the random oracle model. Also, we give a formal security model for group signature and prove that the proposed scheme has the properties of traceability and anonymity.

Keywords: EDL Signature; Group Signature; Pairings; Security Model

1 Introduction

1.1 Background

Group signature [18] allows group member (signer) to hide his identifying information to a group when group member signs messages, thus group signature only reveals the fact that a message was signed by possible one of group members (a list of possible signers). Additionally, in a practical group signature scheme, the group must be constructed by a group manager, who can revoke the anonymity of any signer or identify the real group signer. Because a list of possible signers must be constructed to form a group, some intricate problems need to be solved, such as joining the new members and the revocation of group members. Ateniese *et al.* [1] first proposed an efficient and provably coalition-resistant group signature scheme. However, the security of coalition-resistant group signature was not formalized. In [6], Bellare *et al.* summarized the requirements of group signature and showed the security definitions of group signature. Boneh *et al.* [7] proposed a short group signature scheme in the random oracle model.

In public key cryptography, the management of public keys is a critical problem. For example, certificate authority (CA) generates a digital certificate, which assures that public key belongs to the corresponding user. Then, in a group signature scheme based on public key cryptography, a group public key is corresponding to multi-distributing private keys (signing keys), the joining and revocation of group member is an intricate problem [2, 8, 11, 14]. For large group, it is inefficient to update group public key and distributing private keys when a user joins or exits a group. Bresson *et al.* [11] proposed that the signer may prove that his group certificate does not belong to a list of revoked certificates. However, the length of group signature is proportional to the number of revoked group members. Camenisch *et al.* [14] proposed a different way to handle this problem by using accumulators[1]. However, in some pairing-based accumulators [15, 32], the size of public keys linearly grows with the maximal number of accumulations.

The method of verifier-local revocation was proposed by Brickell in [12]. Boneh *et al.* [8] gave the formal definitions of verifier-local revocation. In this kind of approaches [13,27,30,35], the verifiers receive the revocation list of group members from the authority (such as private key generator) when a signature needs to be verified, and non-revoked group members do not need to update their distributing private keys. So, the length of signature does not depend on the number of revoked group members in this model, and the verifiers only need to perform an additional computing to test that whether the signature was signed by a revoked group member on the revocation list of group members. Of course, this kind of approaches increase the verification cost being proportional to the size

[1] An accumulator is a kind of "hash" function mapping a set of values to a short, constant-size string while allowing to efficiently prove that a specific value was accumulated.

of the revocation list.

In 2009, Nakanishi *et al.* [31] proposed a revocable group signature scheme with constant complexities for signing and verifying. Also, group members do not need to update their distributing private keys. However, the size of public keys linearly grows with the maximal number N of users in their scheme. In 2012, Libert *et al.* [28,29] proposed two group signature schemes based on public key cryptography, which have many useful properties [29]: $O(\log N)$-size group public keys, revocation lists of size $O(r)$ ((r) is the number of revoked users), constant membership certificate size, constant signature size and verification time. However, their schemes need to employ pairing computation.

Additionally, with a rapid development of identity-based cryptography [9, 10, 17, 23], some researchers proposed many identity-based signature schemes in the random oracle model or standard model [5, 16, 23, 24]. So, with these identity-based signature (IBS) schemes, a lot of variants, such as the identity-based ring signature schemes [3, 4, 34], the identity-based group signature schemes [21, 25], *etc.*, have also been proposed. In 2011, Ibraimi *et al.* [25] proposed an identity-based group signature with membership revocation in the standard model. However, their security model is not enough complete for identity-based group signature, some notions are confused. And their scheme is not fully identity-based group signature scheme, the master key of the system is still constructed on public key cryptography. In 2014, Emura *et al.* [21] proposed an γ-hiding revocable group signature scheme in the random oracle model. Because their scheme introduces the notion of attributes, their scheme is enough complex and inefficient.

EDL signature.

The EDL signature [19] and its variant [26] are respectively proposed in 1992 and 1999. Because the computations of the EDL signature do not employ pairings, the efficiency of the schemes is very high. In 2003, Goh *et al.* [22] proved the security of the EDL signature may be reduced to the CDH assumption in the random oracle model. In 2005, Chevallier-Mames [20] further improved the efficiency of the EDL signature by offline/online computation and signature coupon [33], whose security may also be reduced to the CDH assumption in the random oracle model.

1.2 Our Contributions

In this paper, we present a public key-based group signature scheme without pairings under the model of verifier-local revocation. Also, we give the formal security models for group signature. Under our security models, the proposed scheme is proved to have the properties of anonymity and traceability with enough security in the random oracle model. In this paper, our contributions are as follows:

- We present a public key-based (and verifier-local revocation) group signature scheme without pairings, which is based on the modified EDL signature. By modifying the EDL signature from [20, 22], we twice use the modified EDL signature to build a complete group signature scheme: a) we first use the modified EDL signature to construct the partial member private keys when the users join a group; b) we again use the modified EDL signature to generate the valid signatures.

- We present a framework for group signature and show a detailed security model. We introduce the Libert *et al.*'s models [25, 29] to our security model. In our security model, we consider three situations for the security of group signature. Under our security model, the proposed group signature scheme is proved to be secure and has a security reduction to the simple standard assumption (computational Diffie-Hellman assumption) in the random oracle model. So, no poly-time adversary can produce a valid group signature on any messages when the adversary may adaptively be permitted to choose messages after executing group-setup oracle, join-user oracle, revoke-user oracle, signature oracle and trace-user oracle.

- Compared with other group signature schemes proposed by [21, 25, 27, 29, 30], the proposed group signature scheme is not based on pairing computation, and has the constant signing time and signature size (the comparisons of the schemes are given in Section 6).

1.3 Outline

The rest of this paper is organized as follows. In Section 2, we review the bilinear pairings and complexity assumptions on which we build. In Section 3, we show a framework for group signature. In Section 4, we set up the security models for group signature. In Section 5, we propose a group signature scheme under our proposed signature framework. In Section 6, we analyze the efficiency and security of the proposed scheme. Finally, we draw our conclusions in Section 7.

2 Preliminaries

Definition 1. *Computational Diffie-Hellman (CDH) Problem:* Let $\mathbb{G}_1$ be a group of prime order q and g be a generator of $\mathbb{G}_1$; for all $(g, g^a, g^b) \in \mathbb{G}_1$, with $a, b \in \mathbb{Z}_q$, the CDH problem is to compute $g^{a \cdot b}$.

Definition 2. The $(\hbar, \varepsilon)$-CDH assumption holds if no $\hbar$-time algorithm can solve the CDH problem with probability at least ε.

3 A Framework for Group Signature

Definition 3. *Group Signature Scheme:* Let ***GS***=(**System-Setup**, **Generate-Key**, **Group-Setup**, **Join-User**, **Revoke-User**, **Sign**, **Verify**, **Trace-User**) be a group signature scheme. In ***GS***, all algorithms are described as follows:

1) ***System-Setup****: The randomized algorithm run by the trusted authority inputs a security parameter* 1^k*, and then outputs all system parameters* GK *on the security parameter* 1^k*.*

2) ***Generate-Key****: The randomized algorithm run by a group member generates his public/private key pair* (pk_i, sk_i) *with* $i \in \{1, 2.....n\}$*, where* n *is the maximal number of users in a group,* pk_i *is the public key of the group member* i *and* sk_i *is the private key of the group member* i*.*

3) ***Group-Setup****: The randomized algorithm run by the trusted authority inputs* $(GK, Infor \in \{0,1\}^*)$*, and then outputs a group private key* sk_g *to a group manager, where* $Infor$ *is a group public identity information (or* $Infor$ *is seen as the public key of group),* sk_g *is a group private key on the management of the group manager.*

4) ***Join-User****: The randomized algorithm run by the group manager inputs* (GK, sk_g, pk_i)*, and then outputs a member private key* csk_i *to a group member, where* csk_i *is the member private key of the group member and* $i \in \{1, 2.....n\}$*.*

5) ***Revoke-User****: The randomized algorithm run by the group manager inputs* $(GK, sk_g, pk_i, RL_{pk}^t)$*, and then outputs an updated revocation list* RL_{pk}^{t+1}*, where* pk_i *is the public key of the revoked user,* $RL_{pk}^t = \{...(pk_j, \Re_{pk_j})...\}$ *is a revocation list in the duration* t *(*pk_j *is the public key of the revoked user and* $\Re_{pk_j}$ *is a credential on the corresponding public key).*

6) ***Sign****: The randomized algorithm is a standard group signature algorithm. Signer needs to sign a message* $\mathfrak{M} \in \{0,1\}^*$*. The algorithm run by a group member inputs* $(GK, csk_i, \mathfrak{M})$*, and then outputs a signature* σ*, where* $\sigma \in \{0,1\}^* \cup \{\perp\}$*,* csk_i *is the member private key of the group member with* $i \in \{1, 2.....n\}$*.*

7) ***Verify****: The signature receivers verify a standard group signature* σ*. The deterministic algorithm run by a signature verifier inputs* $(GK, \mathfrak{M}, Infor, \sigma, RL_{pk}^t)$*, and then outputs the boolean value, accept or* $reject$*.*

8) ***Trace-User****: The group manager traces a real group member (signer) on group signature* σ*. The deterministic algorithm run by the group manager inputs* $(GK, \mathfrak{M}, Infor, sk_g, \sigma, RL_{pk}^t)$*, and then outputs the corresponding public key of the real signer or* $\perp$*.*

The *correctness* of **GS** requires that for any $GK \leftarrow$***System-Setup***(1^k), $sk_g \leftarrow$ ***Group-Setup***$(GK, Infor \in \{0,1\}^*)$, $csk_i \leftarrow$***Join-User***(GK, sk_g, pk_i) for all i with $i \in \{1, 2.....n\}$, $\mathfrak{M} \in \{0,1\}^*$, then

$$\mathbf{Pr}[\textbf{\textit{Verify}}(GK, \mathfrak{M}, Infor, \textbf{\textit{Sign}}(GK, csk_i, \mathfrak{M}), RL_{pk}^t)=1]=1.$$

The *traceability* of **GS** requires that for any $GK \leftarrow$***System-Setup***(1^k), $sk_g \leftarrow$ ***Group-Setup***$(GK, Infor \in \{0,1\}^*)$, $csk_i \leftarrow$***Join-User***(GK, sk_g, pk_i) for all i with $i \in \{1, 2.....n\}$, $\mathfrak{M} \in \{0,1\}^*$, then

$$\mathbf{Pr}[\textbf{\textit{Trace-User}}(GK, \mathfrak{M}, Infor, sk_g, \textbf{\textit{Sign}}(GK, csk_i, \mathfrak{M}), RL_{pk}^t)=pk_i]=1,$$

where the public key pk_i belongs to the group named by the identity information $Infor$.

4 Security Model

According to [25, 29], we consider that a secure group signature scheme must meet the following three security requirements:

1) **Unforgeability**: A valid group signature must be signed by a valid group member (signer). Therefore, no poly-time adversary can produce a valid group signature on any messages when the adversary may adaptively be permitted to choose messages after executing group setup oracle, joining user oracle, revoking user oracle, signature oracle and tracing user oracle.

2) **Anonymity**: A valid group signature can only reveal that one group identity possessed by a group manager satisfies the signature. It means a valid group signature can hide the identifying information of real signer to one group.

3) **Traceability**: In some situations, a valid group signature needs to reveal the identity (or public key) of real signer from one group. It means a valid group signature can trace a real signer. Then we split the requirement to the following two small security notions[2] [29]:

   a) The first one is called security against *misidentification attacks*, which requires that even if the adversary can introduce (or corrupt) and revoke any user, a valid group signature can not reveal the identifying information outside the set of the identities of unrevoked adversarially-controlled users.

[2]The two security notions are more detailedly expanded from the correctness of traceability.

b) The second one is called security against *framing attacks*, which requires that an honest user is only responsible for the messages that he signed, namely there is no situation that a valid group signature can reveal the identity of a real group member (signer) but this signer did not sign this signature.

Based on the above three situations, we propose a complete security model for group signature. To make our security model easier to understand, we construct several algorithms interacting with adversary, which may make attack experiments to the group signature schemes in the above three situations. In our security model, we maximize adversary's advantage, and assume that all attacking conditions needed by adversary hold and adversary may forge signatures after limitedly querying oracles in the above three situations.

In our security model, we assume there are n users in a group signature scheme ($n \in \mathbb{N}$ is a maximal number of group members), and at least one user u^* of n users is not corrupted by adversary. And we maximize adversary's advantage, where adversary can get all useful information except for the private key of u^*.

All symbols and parameters are defined as follows in the algorithms:

1) U^a is a set of users that were registered by an adversary in this game, where the user $u_i^a \in U^a$ with $i \in \{1, 2.....\}$, pk_i^a is the public key of the user u_i^a.

2) U^b is a set of honest users when an adversary acts a dishonest group manager in this game, where the user $u_i^b \in U^b$ with $i \in \{1, 2.....\}$, pk_i^b is the public key of the user u_i^b.

3) k is a secure parameter, $\mathcal{A}$ represents an adversary.

Definition 4. *Unforgeability of A Group Signature Scheme:* Let ***GS***=(**System-Setup**, **Generate-Key**, **Group-Setup**, **Join-User**, **Revoke-User**, **Sign**, **Verify**, **Trace-User**) be a group signature scheme. Additionally, we set that k is a secure parameter, and $\mathbf{Pr}(\mathcal{B}_{U\_GS}(k,\mathcal{A})=1)$ is the probability that the algorithm $\mathcal{B}_{U\_GS}$ returns 1. Then the advantage that the adversary $\mathcal{A}$ breaks ***GS*** is defined as follows:

$$\mathbf{Adv}_{GS}^{u\_gs-uf}(k, q_g, q_j, q_s, \hbar)=\mathbf{Pr}(\mathcal{B}_{u\_gs}(k,\mathcal{A})=1),$$

where q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_s is the maximal number of "Sign" oracle queries and $\hbar$ is the running time of $\mathcal{B}$. If the advantage that the adversary breaks **GS** *is negligible, then the scheme* **GS** *is secure.*

According to the Definition 4, the algorithm $\mathcal{B}_{U\_GS}$ is described as follows:

Setup: Running ***System-Setup***, $GK \leftarrow$***System-Setup***(1^k), and then GK is passed to $\mathcal{A}$.

Queries: $\mathcal{A}$ makes queries to the following oracles for polynomially many times:

Group-Setup(): Given the public parameters GK and the identity information $Infor$ of the group, the oracle returns a group private key sk_g to $\mathcal{A}$.

Join-User(): Given the public parameters GK, the group private key sk_g (or the identity $Infor$) and the public key pk_i of the group member, the oracle returns a group member private key csk_i to $\mathcal{A}$, where sk_g is a group private key on the identity $Infor$ of the group.

Sign(): Given the public parameters GK, the group member private key csk_i (or the public key pk_i) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$, where $\sigma \in \{0,1\}^* \cup \{\perp\}$.

Forgery: $\mathcal{A}$ outputs its forgery, $(\mathfrak{M}^*, \sigma^*)$ for $Infor^*$ and $RL_{pk^*}^t$, where the identity $Infor^*$ and the revocation list $RL_{pk^*}^t$ are arbitrary forgeries generated by $\mathcal{A}$. It succeeds if

1) $1 \leftarrow$***Verify***$(GK, \mathfrak{M}^*, Infor^*, \sigma^*, RL_{pk^*}^t)$;

2) $\mathcal{A}$ did not query ***Group-Setup*** on input $Infor^*$, did not query ***Join-User*** on inputs sk_g^* and pk^*, and did not query ***Sign*** on inputs csk^* and $\mathfrak{M}^*$, where the public key pk^* belongs to the group named by the identity $Infor^*$.

Definition 5. *Traceability of A Group Signature Scheme:* Let ***GS***=(**System-Setup**, **Generate-Key**, **Group-Setup**, **Join-User**, **Revoke-User**, **Sign**, **Verify**, **Trace-User**) be a group signature scheme, which meets the requirement of unforgeability. ***GS*** is traceable if the following conditions can be satisfied*:*

1) *For all valid generated $GK \leftarrow$**System-Setup**(1^k), $sk_g \leftarrow$**Group-Setup**$(GK, Infor)$, $csk_i \leftarrow$ **Join-User**(GK, sk_g, pk_i) with $i \in \{0,1\}$, then $\sigma_0 =$**Sign**$(GK, csk_0, \mathfrak{M})$ and $\sigma_1 =$**Sign**$(GK, csk_1, \mathfrak{M})$, the outputs of **Trace-User**$(GK, \mathfrak{M}, Infor, sk_g, \sigma_0, RL_{pk}^t)$ and **Trace-User**$(GK, \mathfrak{M}, Infor, sk_g, \sigma_1, RL_{pk}^t)$ are distinguishable in polynomially many times.*

2) *We set that k is a secure parameter, and $\mathbf{Pr}(\mathcal{B}_{TM\_GS}(k,\mathcal{A})=1)$ is the probability that the algorithm $\mathcal{B}_{TM\_GS}$ returns 1, and that $\mathbf{Pr}(\mathcal{B}_{TF\_GS}(k,\mathcal{A})=1)$ is the probability that the algorithm $\mathcal{B}_{TF\_GS}$ returns 1. Then the advantage that the adversary $\mathcal{A}$ breaks* **GS** *is defined as follows:*

$$\mathbf{Adv}_{GS}^{t\_gs-mf}(k, q_g, q_j, q_r, q_s, \hbar)=\mathbf{Pr}(\mathcal{B}_{tm\_gs}(k,\mathcal{A})=1) \| \mathbf{Pr}(\mathcal{B}_{tf\_gs}(k,\mathcal{A})=1),$$

where q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_r is the maximal number of "Revoke-User" oracle queries, q_s is the maximal number of "Sign" oracle queries and $\hbar$ is the running time of $\mathcal{B}$. If the advantage that the adversary breaks **GS** *is negligible, then the scheme* **GS** *is secure.*

According to the Definition 5, the algorithm $\mathcal{B}_{TM\_GS}$ is described as follows:

Setup: Running ***System-Setup***, $GK \leftarrow$***System-Setup***(1^k), and then GK is passed to $\mathcal{A}$.

Queries: $\mathcal{A}$ makes queries to the following oracles for polynomially many times:

Join-User(): Given the public parameters GK, the group private key sk_g (or the identity $Infor$) and the public key $pk_{u_i^a}$ of the group member u_i^a, the oracle returns a group member private key $csk_{u_i^a}$ to $\mathcal{A}$, where sk_g is a group private key on the identity $Infor$ of the group and the user (group member) u_i^a is added to the set U^a.

Revoke-User(): Given the public parameters GK, the group private key sk_g (or the identity $Infor$), the public key $pk_{u_i^a}$ of the revoked group member u_i^a and the revocation list RL_{pk}^t of the last duration t, the oracle returns an updated revocation list RL_{pk}^{t+1}.

Sign(): Given the public parameters GK, the group member private key $csk_{u_i^a}$ (or the public key $pk_{u_i^a}$) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$, where $\sigma \in \{0,1\}^* \cup \{\perp\}$, and the user u_i^a is added to the set U^a if $u_i^a \notin U^a$.

Forgery: $\mathcal{A}$ outputs its forgery, $(\mathfrak{M}^*, \sigma^*)$ for $Infor^*$ and $RL_{pk^*}^t$, where the identity $Infor^*$ and the revocation list $RL_{pk^*}^t$ are arbitrary forgeries generated by $\mathcal{A}$. It succeeds if

1) $1 \leftarrow$***Verify***$(GK, \mathfrak{M}^*, Infor^*, \sigma^*, RL_{pk^*}^t)$;
2) $\mathcal{A}$ did not query ***Join-User*** on inputs sk_g^* and pk^*, did not query ***Revoke-User*** on inputs sk_g^*, pk^* and $RL_{pk^*}^{t-1}$, and did not query ***Sign*** on inputs csk^* and $\mathfrak{M}^*$, where the public key pk^* of the user u_{pk^*} belongs to the group named by the identity $Infor^*$ and $u_{pk^*} \notin U^a \setminus \{u_{pk_i}^a \mid pk_i \in RL_{pk^*}^t\}$;
(c) $pk^* \leftarrow$***Trace-User***$(GK, \mathfrak{M}^*, Infor^*, sk_g^*, \sigma^*, RL_{pk^*}^t)$.

And then the algorithm $\mathcal{B}_{TF\_GS}$ is described as follows:

Setup: Running ***System-Setup***, $GK \leftarrow$***System-Setup***(1^k), and then GK is passed to $\mathcal{A}$.

Queries: $\mathcal{A}$ makes queries to the following oracles for polynomially many times:

Group-Setup(): Given the public parameters GK and the identity $Infor$ of the group, the oracle returns a group private key sk_g to $\mathcal{A}$.

Join-User(): Given the public parameters GK, the group private key sk_g (or the identity $Infor$) and the public key $pk_{u_i^b}$ of the group member u_i^b, the oracle returns a group member private key $csk_{u_i^b}$ to $\mathcal{A}$, where sk_g is a group private key on the identity $Infor$ of the group and the user (group member) u_i^b is added to the set U^b where $U^b \neq \varnothing$.

Revoke-User(): Given the public parameters GK, the group private key sk_g (or the identity $Infor$), the public key $pk_{u_i^b}$ of the revoked group member u_i^b and the revocation list RL_{pk}^t of the last duration t, the oracle returns an updated revocation list RL_{pk}^{t+1}.

Sign(): Given the public parameters GK, the group member private key $csk_{u_i^b}$ (or the public key $pk_{u_i^b}$) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$, where $\sigma \in \{0,1\}^* \cup \{\perp\}$, and the user u_i^b is added to the set U^b if $u_i^b \notin U^b$.

Forgery: $\mathcal{A}$ outputs its forgery, $(\mathfrak{M}^*, \sigma^*)$ for $Infor^*$ and $RL_{pk^*}^t$, where the identity $Infor^*$ and the revocation list $RL_{pk^*}^t$ are arbitrary forgeries generated by $\mathcal{A}$. It succeeds if

1) $1 \leftarrow$***Verify***$(GK, \mathfrak{M}^*, Infor^*, \sigma^*, RL_{pk^*}^t)$;
2) $\mathcal{A}$ did not query ***Group-Setup*** on input $Infor^*$, did not query ***Join-User*** on inputs sk_g^* and pk^*, did not query ***Revoke-User*** on inputs sk_g^*, pk^* and $RL_{pk^*}^{t-1}$, and did not query ***Sign*** on inputs csk^* and $\mathfrak{M}^*$, where the public key pk^* of the user $u_{pk^*}^b$ belongs to the group named by the identity $Infor^*$ and $u_{pk^*}^b \in U^b$;
3) $pk^* \leftarrow$***Trace-User***$(GK, \mathfrak{M}^*, Infor^*, sk_g^*, \sigma^*, RL_{pk^*}^t)$.

Definition 6. *Anonymity of A Group Signature Scheme:* Let ***GS***=(**System-Setup**, **Generate-Key**, **Group-Setup**, **Join-User**, **Revoke-User**, **Sign**, **Verify**, **Trace-User**) be a group signature scheme. Additionally, we set that k is a secure parameter, and $\mathbf{Pr}(\mathcal{B}_{A\_GS}(k,\mathcal{A})=1)$ is the probability that the algorithm $\mathcal{B}_{A\_GS}$ returns 1. Then the advantage that the adversary $\mathcal{A}$ breaks ***GS*** is defined as follows:

$$\mathbf{Adv}_{GS}^{a\_gs}(k, q_g, q_j, q_r, q_s, \hbar) = |\mathbf{Pr}(\mathcal{B}_{a\_gs}(k,\mathcal{A})=1) - \tfrac{1}{2}|,$$

where q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_r is the maximal number of "Revoke-User" oracle queries, q_s is the maximal number of "Sign" oracle queries and $\hbar$ is the running time of $\mathcal{B}$. If the advantage that the adversary breaks **GS** *is negligible, then the scheme* **GS** *is secure.*

According to the Definition 6, the algorithm $\mathcal{B}_{A\_GS}$ is described as follows:

Setup: Running ***System-Setup***, $GK \leftarrow$***System-Setup***(1^k), and then GK is passed to $\mathcal{A}$.

Queries Phase 1: $\mathcal{A}$ makes queries to the following oracles for polynomially many times:

Group-Setup(): Given the public parameters GK and the identity information $Infor$ of the group, the oracle returns a group private key sk_g to $\mathcal{A}$.

Join-User(): Given the public parameters GK, the group private key sk_g (or the identity $Infor$) and the public key pk_i of the group member, the oracle returns a group member private key csk_i to $\mathcal{A}$, where sk_g is a group private key on the identity $Infor$ of the group.

Revoke-User(): Given the public parameters GK, the group private key sk_g (or the identity $Infor$), the public key pk_i of the revoked group member and the revocation list RL_{pk}^{t} of the last duration t, the oracle returns an updated revocation list RL_{pk}^{t+1}.

Sign(): Given the public parameters GK, the group member private key csk_i (or the public key pk_i) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$, where $\sigma \in \{0,1\}^* \cup \{\perp\}$.

Challenge: $\mathcal{A}$ sends to the challenger its forgeries ($\mathfrak{M}^*$, $Infor^*$, $RL_{pk^*}^{t}$) and two group member public keys pk_0^* and pk_1^* that belong to the group named by the group identity $Infor^*$. The forgeries satisfy the following conditions:

1) $\mathcal{A}$ did not query ***Group-Setup*** on input $Infor^*$;
2) $\mathcal{A}$ did not query ***Join-User*** on inputs $Infor^*$, pk_0^* (and pk_1^*);
3) $\mathcal{A}$ did not query ***Revoke-User*** on inputs $Infor^*$, pk_0^* (and pk_1^*) and $RL_{pk^*}^{t-1}$.

The challenger picks a random bit $x \in \{0,1\}$, and then runs and outputs $\sigma^* \leftarrow$***Sign***$(GK, csk_x^*, \mathfrak{M}^*)$ to $\mathcal{A}$.

Queries Phase 2: $\mathcal{A}$ makes queries to the following oracles for polynomially many times again:

Group-Setup(): Given the public parameters GK and the identity information $Infor$ of the group (where $Infor \neq Infor^*$), the oracle returns a group private key sk_g to $\mathcal{A}$.

Join-User(): Given the public parameters GK, the group private key sk_g (or the identity $Infor$) and the public key pk_i of the group member (where $sk_g \neq sk_g^*$ and $pk_i \notin \{pk_0^*, pk_1^*\}$), the oracle returns a group member private key csk_i to $\mathcal{A}$, where sk_g is a group private key on the identity $Infor$ of the group.

Revoke-User(): Given the public parameters GK, the group private key sk_g (or the identity $Infor$), the public key pk_i of the revoked group member and the revocation list RL_{pk}^{t} of the last duration t, the oracle returns an updated revocation list RL_{pk}^{t+1} (where $\mathcal{A}$ did not query ***Revoke-User*** on inputs sk_g^*, pk_0^* (and pk_1^*)).

Sign(): Given the public parameters GK, the group member private key csk_i (or the public key pk_i) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$, where $\sigma \in \{0,1\}^* \cup \{\perp\}$.

Guess: $\mathcal{A}$ outputs a bit $x' \in \{0,1\}$ and succeeds if $x' = x$.

5 Group Signature Scheme Based on EDL Signature

Let **GS**=(***System-Setup***, ***Generate-Key***, ***Group-Setup***, ***Join-User***, ***Revoke-User***, ***Sign***, ***Verify***, ***Trace-User***) be a group signature scheme. In **GS**, all algorithms are described as follows:

GS.*System-Setup*: The algorithm run by the trusted authority inputs a security parameter 1^k. Then, let $\mathbb{G}_1$ be group of prime order q and module p, and g be a generator of $\mathbb{G}_1$. The size of the group is determined by the security parameter. And four hash functions, $H_0 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_1 : \mathbb{G}_1 \to \mathbb{G}_1$, $H_2 : \mathbb{G}_1^4 \times \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_3 : \mathbb{G}_1^3 \times \{0,1\}^* \to \mathbb{Z}_q^*$ can be defined. Finally, the algorithm outputs the public parameters GK=($\mathbb{G}_1$, g, H_0, H_1, H_2, H_3).

GS.*Generate-Key*: The algorithm run by a group member generates his public/private key pair (pk_l, sk_l) with $l \in \{1, 2.....n\}$, where n is the maximal number of users in a group. The algorithm randomly chooses $sk_l \in \mathbb{Z}_q^*$, and then computes $pk_l = g^{sk_l}$.

GS.*Group-Setup*: The algorithm run by the trusted authority inputs (GK, $Infor \in \{0,1\}^*$), where $Infor$ is a group public identity information. The algorithm randomly chooses $d \in \mathbb{Z}_q^*$, computes and outputs a group private key $sk_g = d \cdot H_0(Infor)$ to a group manager, and then publishes the group public key $pk_g = g^d$.

GS.*Join-User*: The algorithm run by the group manager inputs (GK, sk_g, pk_l), and then the following steps are finished:

1) The algorithm run by the group manager randomly chooses $a \in \mathbb{Z}_q^*$, computes

$$\begin{aligned} u_1 &= g^a, \\ h_1 &= H_1(u_1), \\ x_1 &= h_1^{sk_g}, \\ v_1 &= h_1^a, \\ c_1 &= H_2(u_1, x_1, v_1, pk_g, Infor), \\ r &= a + c_1 \cdot sk_g. \end{aligned}$$

The algorithm outputs a partial member private key $\delta = (x_1, c_1, r)$ to a group member whose public key is pk_l, and then saves the tuple (pk_l, u_1), where u_1 is used to trace the real signer.

2) The algorithm run by a group member with the public key pk_l and the private key sk_l verifies the partial member private key $\delta = (x_1, c_1, r)$ by the following computations:

$$\begin{aligned} u_1' &= g^r \cdot (pk_g)^{-c_1 \cdot H_0(Infor)}, \\ h_1' &= H_1(u_1'), \\ v_1' &= (h_1')^r \cdot (x_1)^{-c_1}, \\ c_1' &= H_2(u_1', x_1, v_1', pk_g, Infor), \end{aligned}$$

and then checks $c_1' = c_1$. If the equation $c_1' = c_1$ is correct, the group member accepts δ, otherwise the group member requires that the group manager must resend δ. Finally, the algorithm computes and outputs the group member private key $csk_l = \{u_1', \delta = (x_1, c_1, r)\}$ to the group member, where $u_1' = u_1 = g^a$.

GS.*Revoke-User*: The algorithm run by the group manager inputs (GK, sk_g, pk_l, RL_{pk}^t), where pk_l is the public key of the revoked user. The algorithm computes $rv_l = (pk_l)^{\frac{1}{c_1}}$, where rv_l is a credential on the corresponding public key pk_l. Finally, the algorithm outputs and adds a tuple $[pk_l, rv_l]$ to the revocation list RL_{pk}^t, and then an updated revocation list RL_{pk}^{t+1} is published by a secure approach.

GS.*Sign*: A group member with the group member private key csk_l needs to sign a message $\mathfrak{M} \in \{0,1\}^*$. The algorithm run by the group member inputs (GK, csk_l, $\mathfrak{M}$), and then randomly chooses $k, f \in \mathbb{Z}_q^*$, computes[3]

$$\begin{aligned} u_2 &= g^k \cdot (u_1')^f, \\ h_2 &= H_1(u_2), \\ v_2 &= h_2^{f \cdot r + k}, \\ c_1'' &= c_1 \cdot f, \\ c_2 &= H_3(u_2, v_2, pk_g, c_1'', \mathfrak{M}, Infor), \\ y &= f \cdot r + c_2 \cdot f \cdot sk_l, \\ x_2 &= sk_l \cdot f - \frac{k}{c_2}, x_3 = g^k, x_4 = g^{sk_l \cdot f}. \end{aligned}$$

Finally, the algorithm outputs a signature $\sigma = \{c_1'', c_2, x_2, x_3, x_4, y\}$.

GS.*Verify*: The signature receivers verify a group signature σ. The algorithm run by a signature verifier inputs (GK, $\mathfrak{M}$, $Infor$, σ, RL_{pk}^t), and then the following steps are finished:

1) The algorithm computes the following equations:

$$\begin{aligned} u_2' &= g^y \cdot (pk_g)^{-c_1'' \cdot H_0(Infor)} \cdot g^{-x_2 \cdot c_2}, \\ h_2' &= H_1(u_2'), \\ v_2' &= (h_2')^y \cdot (h_2')^{-x_2 \cdot c_2}, \\ c_2' &= H_3(u_2', v_2', pk_g, c_1'', \mathfrak{M}, Infor), \end{aligned}$$

and then checks $c_2' = c_2$. If the equation $c_2' = c_2$ is correct, then the algorithm runs into the next step, otherwise the algorithm outputs the boolean value $reject$.

2) The algorithm finishes the following steps on the revocation list RL_{pk}^t:

- Check the equation $g^{x_2} = (x_3)^{-\frac{1}{c_2}} \cdot x_4$; if the equation is correct, then the algorithm continues, otherwise the algorithm outputs the boolean value $reject$;
- Compute the equation $u_2'' = g^y \cdot (pk_g)^{-c_1'' \cdot H_0(Infor)} \cdot x_3 \cdot (x_4)^{-c_2}$, then check the equation $u_2'' = u_2'$; if the equation is correct, then the algorithm continues, otherwise the algorithm outputs the boolean value $reject$;
- Compute $rv_l' = (rv_l)^{c_1'' \cdot c_2} = (pk_l)^{\frac{1}{c_1} \cdot c_1 \cdot f \cdot c_2} = (pk_l)^{c_2 \cdot f} = g^{sk_l \cdot c_2 \cdot f}$, and $rv_l'' = g^{x_2 \cdot c_2} \cdot x_3 = g^{sk_l \cdot f \cdot c_2 - k} \cdot x_3 = g^{sk_l \cdot c_2 \cdot f}$, and then check $rv_l' = rv_l''$; if the equation $rv_l' = rv_l''$ is correct, then the algorithm directly outputs the boolean value $reject$; otherwise, if the algorithm does not find the correcting equation $rv_l' = rv_l''$ on the revocation list RL_{pk}^t, then the algorithm outputs the boolean value $accept$.

Remark: $rv_l' = rv_l''$ can denote whether the group member (signer) has been revoked.

GS.*Trace-User*: The group manager traces a real group member (signer) on group signature σ, which can be verified by **GS.*Verify***. The algorithm run by the group manager computes the following equation:

$$\begin{aligned} [\frac{g^{c_1 \cdot (y - x_2 \cdot c_2)}}{(pk_g)^{c_1'' \cdot c_1} \cdot (x_3)^{c_1}}]^{\frac{1}{c_1''}} &= [\frac{g^{c_1 \cdot (f \cdot r + k)}}{(pk_g)^{c_1'' \cdot c_1} \cdot (x_3)^{c_1}}]^{\frac{1}{c_1''}} \\ &= [\frac{g^{c_1 \cdot f \cdot (a + c_1 \cdot sk_g) + c_1 \cdot k}}{(pk_g)^{c_1'' \cdot c_1} \cdot (x_3)^{c_1}}]^{\frac{1}{c_1''}} \\ &= [\frac{g^{c_1'' \cdot a} \cdot g^{sk_g \cdot c_1'' \cdot c_1} \cdot g^{k \cdot c_1}}{(pk_g)^{c_1'' \cdot c_1} \cdot (x_3)^{c_1}}]^{\frac{1}{c_1''}} \\ &= g^a = u_1. \end{aligned}$$

Finally, the algorithm finds and outputs the corresponding public key pk_l by u_1.

[3] c_1'' may be also seen as $\{0,1\}^*$ in the computation of $H_3()$.

6 Analysis of the Proposed Scheme

6.1 Efficiency

In the proposed scheme, $\sigma = \{c_1'', c_2, x_2, x_3, x_4, y\}$, where

$$\begin{aligned} c_1'' &= c_1 \cdot f, \\ c_2 &= H_3(u_2, v_2, pk_g, c_1'', \mathfrak{M}, Infor), \\ y &= f \cdot r + c_2 \cdot f \cdot sk_l, \\ x_2 &= sk_l \cdot f - \frac{k}{c_2}, \\ x_3 &= g^k and x_4 = g^{sk_l \cdot f}. \end{aligned}$$

Thus, the length of signature is $2 \cdot |\mathbb{G}_1| + 4 \cdot |\mathbb{Z}_q^*|$, where $|\mathbb{G}_1|$ is the size of element in $\mathbb{G}_1$ and $|\mathbb{Z}_q^*|$ is the size of element in $\mathbb{Z}_q^*$. Additionally, the signing and verifying procedure is mainly based on integer multiplication and hash computation, so if we assume that the time for integer multiplication and hash computation can be ignored, then signing a message for a group signature only needs to compute 5 exponentiations in $\mathbb{G}_1$ and 1 multiplication in $\mathbb{G}_1$, and verification requires at most $2 \cdot L_r + 8$ exponentiations in $\mathbb{G}_1$ and $L_r + 6$ multiplications in $\mathbb{G}_1$, where L_r is the number of the revoked users in the revocation list RL_{pk}^t[4].

In this paper, we compare the proposed scheme (the scheme of Section 5) with the other group signature schemes [21,25,27,29,30]. Table 1 shows the comparisons of the schemes. Compared with other schemes, although our scheme is constructed in the random oracle model, our scheme does not employ pairing computation and has the constant signing time and signature size.

6.2 Security

In the section, we show the proposed scheme (the scheme of Section 5) has the unforgeability, traceability and anonymity under the adaptive chosen message attacks, which can be reduced to the CDH assumption. Our proofs for the following theorems are based on the security models of Section 4[5].

Theorem 1. The scheme of Section 5 is ($\hbar$, ε, q_g, q_j, q_s)-unforgeable (according to the Definition 4), assuming that the ($\hbar'$, ε')-CDH assumption holds in $\mathbb{G}_1$, where:

$$\begin{aligned} \varepsilon' &= \varepsilon - \frac{q_g}{2^{n_q}} - q_j \cdot (\frac{1}{2^{n_q}} + \frac{2 \cdot q_h}{2^{n_q}}) \\ &\quad - \frac{q_s \cdot q_h}{2^{6 \cdot n_q}} - \frac{q_s \cdot (q_h + q_s)}{2^{n_q}}, \\ \hbar' &= \hbar + O((q_h + q_g + 4 \cdot q_j + 12 \cdot q_s) \cdot C_{exp} \\ &\quad + 4 \cdot q_s \cdot C_{mul}), \end{aligned}$$

and q_h is the maximal number of "Hash" oracle queries, q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_s is the maximal number of "Sign" oracle queries, C_{mul} and C_{exp} are respectively the time for a multiplication and an exponentiation in $\mathbb{G}_1$.

Proof. Let **GS** be a group signature scheme of Section 5. Additionally, let $\mathcal{A}$ be an ($\hbar$, ε, q_g, q_j, q_s)-adversary attacking **GS**.

From the adversary $\mathcal{A}$, we construct an algorithm $\mathcal{B}$, for (g, g^a, g^b)$\in \mathbb{G}_1$, the algorithm $\mathcal{B}$ is able to use $\mathcal{A}$ to compute $g^{a \cdot b}$. Thus, we assume the algorithm $\mathcal{B}$ can solve the CDH with probability at least ε' and in time at most $\hbar'$, contradicting the ($\hbar'$, ε')-CDH assumption. Such a simulation may be created in the following way:

Setup: The trusted authority system inputs a security parameter 1^k. Then, let $\mathbb{G}_1$ be group of prime order q and module p, and g be a generator of $\mathbb{G}_1$. The size of the group is determined by the security parameter. Also, $H_0 : \{0,1\}^* \to \mathbb{Z}_q^*$ can directly be computed on no querying. $H_1 : \mathbb{G}_1 \to \mathbb{G}_1$, $H_2 : \mathbb{G}_1^4 \times \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_3 : \mathbb{G}_1^3 \times \{0,1\}^* \to \mathbb{Z}_q^*$ can be simulated by the algorithms H_1 Queries, H_2 Queries and H_3 Queries, where we set that g^b ($\mathcal{B}$ does not know b) is used to answer the query on H_1 Queries. Additionally, we assume that the user u^* is a challenger, whose public key is $pk^* = g^a$ ($\mathcal{B}$ does not know a where a is seen as the corresponding private key). Finally, the algorithm outputs the public parameters GK=($\mathbb{G}_1$, g, H_0).

Queries: When running the adversary $\mathcal{A}$, the relevant queries can occur according to the Definition 4. The algorithm $\mathcal{B}$ answers these in the following way:

H_1 Queries: If this query is fresh, then the algorithm chooses random $s \in Z_q^*$, computes and outputs $(g^b)^s = g^{b \cdot s}$ to the adversary $\mathcal{A}$; otherwise the algorithm returns the same result. Also, the algorithm saves the new tuple $(s, g^{b \cdot s})$ to U_List.

H_2 Queries: If this query is fresh, then the algorithm outputs the new result to the adversary $\mathcal{A}$; otherwise the algorithm returns the same result.

H_3 Queries: If this query is fresh, then the algorithm outputs the new result to the adversary $\mathcal{A}$; otherwise the algorithm returns the same result.

Group-Setup Queries: Given the public parameters GK and the identity information $Infor$ of the group, the algorithm randomly chooses $d \in \mathbb{Z}_q^*$, computes and outputs a group private key $sk_g = d \cdot H_0(Infor)$ and a group public key $pk_g = g^d$ to $\mathcal{A}$.

[4] We only consider the bad thing that the revoked user is the last one in the revocation list when verification starts from the first one to the last one.

[5] As the proofs of Theorem 2 and Theorem 3 are similar to the proof of Theorem 1, we omit the similar proofs in this paper.

Table 1: Comparisons of the six schemes

	Signature Size	Signature Cost	Verification Cost	Model
Scheme [30]	$O(1)$	$O(1)$	$O(L_r)$	random oracle
Scheme [27]	$O(1)$	$O(1)$	$O(L_r)$	without random oracle
Scheme [29]	$O(1)$	$O(1)$	$O(1)$	without random oracle
Scheme [25]	$O(1)$	$O(L_m)$	$O(L_m + L_k)$	without random oracle
Scheme [21]	$O(1)$	$O(L_m)$	$O(1)$	random oracle
Our Scheme	$O(1)$	$O(1)$	$O(L_r)$	random oracle

caption: L_m is the length of signed message, L_k is the length of user identity, L_r is the number of revoked users in the revocation list.

Join-User Queries: Given the public parameters GK and the group identity $Infor$, the algorithm randomly chooses $t, d \in \mathbb{Z}_q^*$, computes

$$\begin{aligned} u_1 &= g^t, h_1 = H_1(u_1), \\ x_1 &= h_1^{d \cdot H_0(Infor)}, \\ v_1 &= h_1^t, \\ c_1 &= H_2(u_1, x_1, v_1, g^d, Infor), \\ r &= t + c_1 \cdot d \cdot H_0(Infor). \end{aligned}$$

The algorithm outputs a partial member private key $\delta = (x_1, c_1, r)$ to $\mathcal{A}$. Because the algorithm does not know the private key of the queried group member, the algorithm only outputs a partial member private key to $\mathcal{A}$. However, the adversary $\mathcal{A}$ is easy to compute out the complete group member private key when the adversary $\mathcal{A}$ corrupted some group members or registered some controlled group member to the simulation system.

Sign Queries: Given the public parameters GK, the identity information $Infor$ of the group, the public key pk_l and the message $\mathfrak{M}$, the following setups are finished:

1) The algorithm randomly chooses $t, d \in \mathbb{Z}_q^*$, computes

$$\begin{aligned} u_1 &= g^t, \\ h_1 &= H_1(u_1), \\ x_1 &= h_1^{d \cdot H_0(Infor)}, \\ v_1 &= h_1^t, \\ c_1 &= H_2(u_1, x_1, v_1, g^d, Infor). \end{aligned}$$

2) The algorithm randomly chooses $c_2, y, f, k \in \mathbb{Z}_q^*$, computes
$$u_2 = g^y \cdot g^{-d \cdot c_1 \cdot f \cdot H_0(Infor)} \cdot g^{-k},$$
and then queries the oracle **H_1 Queries** for u_2, if u_2 has been queried, then the algorithm aborts; otherwise the algorithm continues.
3) The algorithm randomly chooses $j \in \mathbb{Z}_q^*$, computes
$$v_2 = h_2^y \cdot g^{-k \cdot j},$$
where we set $h_2 = H_1(u_2) = g^j$ (satisfy the condition that $DL_{h_2}((h_2)^k) = DL_g(g^k) = k$).
4) The algorithm queries the oracle **H_3 Queries**, if the tuple $(u_2, v_2, g^d, c_1 \cdot f, \mathfrak{M}, Infor)$ has been queried, then the algorithm aborts; otherwise the algorithm continues.
5) The algorithm computes $x_2 = \frac{k}{c_2}$, $x_3 = g^{-k} \cdot (pk_l)^f$, $x_4 = (pk_l)^{\frac{f}{c_2}}$, and then outputs a group signature $\sigma = \{c_1', c_2, x_2, x_3, x_4, y\}$ to the adversary $\mathcal{A}$, and saves the tuple (t, d, c_2, f, k) to S_List.

Forgery: If the algorithm $\mathcal{B}$ does not abort as a consequence of one of the queries above, the adversary $\mathcal{A}$ will, with probability at least ε, return a forgery $(\mathfrak{M}^*, \sigma^*, Infor^*, RL_{pk^*}^t)$ for the challenger u^*, where the identity $Infor^*$ and the revocation list $RL_{pk^*}^t$ are arbitrary forgeries generated by $\mathcal{A}$. And the forgery satisfies the following condition:

1) $1 \leftarrow$ ***Verify***$(GK, \mathfrak{M}^*, Infor^*, \sigma^*, RL_{pk^*}^t)$;
2) $\mathcal{A}$ did not query ***Group-Setup*** on input $Infor^*$, did not query ***Join-User*** on input $Infor^*$, and did not query ***Sign*** on inputs $Infor^*$, pk^* and $\mathfrak{M}^*$ where the public key pk^* of the challenger u^* belongs to the group named by the identity $Infor^*$.

Then, if the adversary $\mathcal{A}$ did not query the oracle **H_1 Queries**, or U_List is empty or S_List is empty, then the algorithm $\mathcal{B}$ aborts.

Otherwise, the algorithm $\mathcal{B}$ can get $h_2 = H_1(*) = g^{b \cdot s}$. So, when the condition $DL_{h_2}((h_2)^{a \cdot f \cdot c_2 - k}) = DL_g(g^{a \cdot f \cdot c_2 - k}) = a \cdot f \cdot c_2 - k$ holds, we can get the followings:

$$\begin{aligned} h_2^{x_2 \cdot c_2} &= (h_2)^{(a \cdot f - \frac{k}{c_2}) \cdot c_2} \\ &= (g^{b \cdot s})^{(a \cdot f - \frac{k}{c_2}) \cdot c_2} \\ &= (g^{b \cdot s})^{(a \cdot f \cdot c_2 - k)} \\ &= g^{a \cdot b \cdot s \cdot f \cdot c_2 - b \cdot s \cdot k}, \end{aligned}$$

then $\mathcal{B}$ computes and outputs $(h_2^{x_2 \cdot c_2} \cdot g^{b \cdot s \cdot k})^{\frac{1}{c_2 \cdot s \cdot f}} = g^{a \cdot b}$, which is the solution to the given CDH problem.

Now, we analyze the probability of the algorithm $\mathcal{B}$ not aborting. For the simulation to complete without aborting, we require that all ***Group-Setup*** queries and all ***Join-User*** queries are fresh, and all ***Sign*** queries do not abort. So, if the algorithm $\mathcal{B}$ does not abort, then the following conditions must hold:

1) All ***Group-Setup*** queries are fresh, because $H_0 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ is uniformly distributed in $\mathbb{Z}_q$, the collision probability of H_0 is $\frac{1}{2^{n_q}}$, then the failure probability of the queries is at most $\frac{q_g}{2^{n_q}}$.
2) All ***Join-User*** queries are fresh, similarly the collision probability of H_0 is $\frac{1}{2^{n_q}}$, and because $t, d \in \mathbb{Z}_q^*$ are uniformly distributed in $\mathbb{Z}_q$, the collision probability of H_1 is $q_h \cdot \frac{1}{2^{n_q}} = \frac{q_h}{2^{n_q}}$ and the collision probability of H_2 is $q_h \cdot \frac{1}{2^{n_q}} = \frac{q_h}{2^{n_q}}$, then the failure probability of the queries is at most $q_j \cdot (\frac{1}{2^{n_q}} + \frac{2 \cdot q_h}{2^{n_q}})$.
3) All ***Sign*** queries do not abort, then we may get the followings:
   - The algorithm may abort in the setup b), namely u_2 has been queried on the oracle **H_1 Queries**. So, as $t, d, c_2, y, f, k \in \mathbb{Z}_q^*$ are uniformly distributed in $\mathbb{Z}_q^6$, the collision probability of H_1 is $q_h \cdot \frac{1}{2^{6 \cdot n_q}} = \frac{q_h}{2^{6 \cdot n_q}}$, then the failure probability of the queries is at most $\frac{q_s \cdot q_h}{2^{6 \cdot n_q}}$;
   - The algorithm may abort in the setup d), namely the tuple $(u_2, v_2, g^d, c_1 \cdot f, \mathfrak{M}, Infor)$ has been queried on the oracle **H_3 Queries**. So, as $j \in \mathbb{Z}_q^*$ is uniformly distributed in $\mathbb{Z}_q$, the collision probability of H_3 is $(q_h + q_s) \cdot \frac{1}{2^{n_q}} = \frac{q_h + q_s}{2^{n_q}}$, then the failure probability of the queries is at most $\frac{q_s \cdot (q_h + q_s)}{2^{n_q}}$.

Therefore, from the above analysis, we get that the algorithm $\mathcal{B}$ can compute $g^{a \cdot b}$ from the forgery as shown above, with probability at least $\varepsilon' = \varepsilon - \frac{q_g}{2^{n_q}} - q_j \cdot (\frac{1}{2^{n_q}} + \frac{2 \cdot q_h}{2^{n_q}}) - \frac{q_s \cdot q_h}{2^{6 \cdot n_q}} - \frac{q_s \cdot (q_h + q_s)}{2^{n_q}}$. The time complexity of the algorithm $\mathcal{B}$ is $\hbar' = \hbar + O((q_h + q_g + 4 \cdot q_j + 12 \cdot q_s) \cdot C_{exp} + 4 \cdot q_s \cdot C_{mul})$, where we assume that the time for integer addition, integer multiplication and hash computation can both be ignored.

Thus, Theorem 1 follows. □

Theorem 2. The scheme of Section 5 is a traceable group signature scheme when it is unforgeable (Theorem 1 holds) and satisfies the following conditions (according to the Definition 5):

1) *The outputs of "Trace-User" oracle are distinguishable in polynomially many times*;
2) *The scheme of Section 5 is $(\hbar'', \varepsilon'', q_g, q_j, q_r, q_s)$-secure, assuming that the $(\hbar', \varepsilon')$-CDH assumption holds in $\mathbb{G}_1$, where:*

$$\begin{aligned}
\varepsilon'' &= [\varepsilon' + q_j \cdot (\frac{1}{2^{n_q}} + \frac{2 \cdot q_h}{2^{n_q}}) \\
&\quad + q_r \cdot (\frac{1}{2^{n_q}} + \frac{2 \cdot q_h}{2^{n_q}}) \\
&\quad + \frac{q_s \cdot q_h}{2^{6 \cdot n_q}} + \frac{q_s \cdot (q_h + q_s)}{2^{n_q}}] \\
&\quad \| [\varepsilon' + \frac{q_g}{2^{n_q}} + q_j \cdot (\frac{1}{2^{n_q}} + \frac{2 \cdot q_h}{2^{n_q}}) \\
&\quad + q_r \cdot (\frac{1}{2^{n_q}} + \frac{2 \cdot q_h}{2^{n_q}}) + \frac{q_s \cdot q_h}{2^{6 \cdot n_q}} + \frac{q_s \cdot (q_h + q_s)}{2^{n_q}}], \\
\hbar'' &= \mathbf{MAX}\{\hbar' - O((q_h + 4 \cdot q_j + 5 \cdot q_r + 12 \cdot q_s) C_{exp} \\
&\quad + 4 \cdot q_s \cdot C_{mul}), \quad \hbar' - O((q_h + g_g + 4 \cdot q_j \\
&\quad + 5 \cdot q_r + 12 \cdot q_s) \cdot C_{exp} + 4 \cdot q_s \cdot C_{mul})\}.
\end{aligned}$$

and q_h is the maximal number of "Hash" oracle queries, q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_r is the maximal number of "Revoke-User" oracle queries, q_s is the maximal number of "Sign" oracle queries, C_{mul} and C_{exp} are respectively the time for a multiplication and an exponentiation in $\mathbb{G}_1$.

Theorem 3. The scheme of Section 5 is $(\hbar, \varepsilon, q_g, q_j, q_r, q_s)$-anonymous (according to the Definition 6), assuming that the $(\hbar', \varepsilon')$-CDH assumption holds in $\mathbb{G}_1$, where:

$$\begin{aligned}
\varepsilon' &= \varepsilon - \frac{q_{g_1} + q_{g_2}}{2^{n_q}} - (q_{j_1} + q_{j_2}) \cdot (\frac{1}{2^{n_q}} + \frac{2 \cdot q_h}{2^{n_q}}) \\
&\quad - (q_{r_1} + q_{r_2}) \cdot (\frac{1}{2^{n_q}} + \frac{2 \cdot q_h}{2^{n_q}}) - \frac{(q_{s_1} + q_{s_2}) \cdot q_h}{2^{6 \cdot n_q}} \\
&\quad - \frac{(q_{s_1} + q_{s_2}) \cdot (2 \cdot q_h + q_{s_1} + q_{s_2})}{2^{n_q}}, \\
\hbar' &= \hbar + O((q_h + q_{g_1} + q_{g_2} + 4 \cdot (q_{j_1} + q_{j_2}) \\
&\quad + 5 \cdot (q_{r_1} + q_{r_2}) + 12 \cdot (q_{s_1} + q_{s_2})) \cdot C_{exp} \\
&\quad + 4 \cdot (q_{s_1} + q_{s_2}) \cdot C_{mul}),
\end{aligned}$$

and q_h is the maximal number of "Hash" oracle queries, q_{g_1} and q_{g_2} are respectively the maximal numbers of "Group-Setup" oracle queries in the Queries Phase 1 and 2, q_{j_1} and q_{j_2} are respectively the maximal numbers of "Join-User" oracle queries in the Queries Phase 1 and 2, q_{r_1} and q_{r_2} are respectively the maximal numbers of "Revoke-User" oracle queries in the Queries Phase 1 and 2, q_{s_1} and q_{s_2} are respectively the maximal numbers of "Sign" oracle queries in the Queries Phase 1 and 2, C_{mul} and C_{exp} are respectively the time for a multiplication and an exponentiation in $\mathbb{G}_1$.

7 Conclusions

In this paper, by modifying the EDL signature, we present a public key-based group signature scheme in the random

oracle, which is based on the model of verifier-local revocation. Also, we give the security models for group signature. Under our security models, the proposed scheme is proved to have the properties of anonymity and traceability with enough security. Compared with other group signature schemes proposed by [21,25,27,29,30], the proposed group signature scheme does not employ pairing computation and has the constant signature size, so the proposed scheme is efficient. However, because the proposed scheme is not enough efficient in revoking verification of signatures, the work about group signature still needs to be further progressed.

Acknowledgments

This study is funded by the Open Research Fund of Key Laboratory of Network Crime Investigation of Hunan Provincial Colleges (No.2017WLFZZC003), the National Natural Science Foundations of China (No.61402055, No.61504013) and the Hunan Provincial Natural Science Foundation of China (No.2018JJ2445, No.2016JJ3012). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] G. Ateniese, J. Camenisch, M. Joye, G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Annual International Cryptology Conference*, pp. 255-270, 2000.

[2] G. Ateniese, D. Song, G. Tsudik, "Quasi-efficient revocation in group signatures," in Proceedings of the 6th International Conference on Financial Cryptography, pp. 183-197, 2002.

[3] M. H. Au, J. K. Liu, W. Susilo, T. H. Yuen, "Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction," *Theoretical Computer Science*, vol. 469, pp. 1-14, 2013.

[4] A. K. Awasthi, S. Lal, "ID-based ring signature and proxy ring signature schemes from bilinear pairings," *International Journal of Network Security*, vol. 4, no. 2, pp. 187-192, 2007.

[5] P. S. L. M. Barreto, B. Libert, N. McCullagh, J. Quisquater, "Efficient and provably-secure identity-Based signatures and signcryption from bilinear maps," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 515-532, 2005.

[6] M. Bellare, D. Micciancio, B. Warinschi, "Foundations of group signatures: Formal definitions, simplified require- ments, and a construction based on general assumptions," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 614-629, 2003.

[7] D. Boneh, X. Boyen, H. Shacham, "Short group signatures," in *Annual International Cryptology Conference*, pp. 41-55, 2004.

[8] D. Boneh, H. Shacham, "Group signatures with verifier-local revocation," in *Proceedings of the 11th ACM conference on Computer and Communications Security*, pp. 168-177, 2004.

[9] D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing," in *Annual International Cryptology Conference*, 213-229, 2001.

[10] D. Boneh, M. Hanburg, "Generalized identity based and broadcast encryption schemes," in *International Conference on the Theory and Application of Cryptology and Information Security*, 455-470, 2008.

[11] E. Bresson, J. Stern, "Efficient revocation in group signatures," in *International Workshop on Public Key Cryptography*, pp. 190-206, 2001.

[12] E. Brickell, "An efficient protocol for anonymously providing assurance of the container of the private key," *Submission to the Trusted Computing Group*, 2003. (`https://www.semanticscholar.org/paper/An-e-cient-protocol-for-anonymously-providing-of-of-Brickell/0a780b09cdcee20cc617d5b840f4c9dafb398fa8`)

[13] E. Brickell, J. Camenisch, L. Chen, "Direct anonymous attestation," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pp. 132-145, 2004.

[14] J. Camenisch, A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Annual International Cryptology Conference*, pp. 61-76, 2002.

[15] J. Camenisch, M. Kohlweiss, C. Soriente, "An accumulator based on bilinear maps and efficient revocation for anonymous credentials," in *International Workshop on Public Key Cryptography*, pp. 481-500, 2009.

[16] J. C. Cha, J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *International Workshop on Public Key Cryptography*, pp. 18-30, 2002.

[17] C. C. Chang, C. Y. Sun, 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.

[18] D. Chaum, E. van Heyst, "Group signatures," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 257-265, 1991.

[19] D. Chaum, T. P. Pedersen, "Wallet databases with observers," in *Annual International Cryptology Conference*, pp. 89-105, 1992.

[20] B. C. Mames, "An efficient CDH-based signature scheme with a tight security reduction," in *Annual International Cryptology Conference*, pp. 511-526, 2005.

[21] K. Emura, A. Miyaji, K. Omote, "An r-hiding revocable group signature scheme: Group signatures with the property of hiding the number of revoked users," *Journal of Applied Mathematics*, vol. 2014, no. 272, pp. 14, 2014.

[22] E. J. Goh, S. Jarecki, "A signature scheme as secure as the Diffie-Hellman problem," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 401-415, 2003.

[23] D. He, M. K. Khan, S. Wu, "On the security of a RSA-based certificateless signature scheme," *International Journal of Network Security*, vol. 16, no. 1, pp. 78-80, 2014.

[24] F. Hess, "Efficient identity based signature schemes based on pairings," in *International Workshop on Selected Areas in Cryptography*, 310-324, 2002.

[25] L. Ibraimi, S. Nikova, P. Hartel, W. Jonker, *An Identity-Based Group Signature with Membership Revocation in the Standard Model*, 2010. (`https://pdfs.semanticscholar.org/7a1c/0f61d15c957d3c599779f2aafbca0ae1eae8.pdf`)

[26] M. Jakobsson, C. Schnorr, "Efficient oblivious proofs of correct exponentiation," *Secure Information Networks*, volume 23, pp. 71-86, 1999.

[27] B. Libert, D. Vergnaud, "Group signatures with verifier-local revocation and backward unlinkability in the standard model," in *International Conference on Cryptology and Network Security*, pp. 498-517, 2009.

[28] B. Libert, T. Peters, M. Yung, "Scalable group signatures with revocation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp.609-627, 2012.

[29] B. Libert, T. Peters, M. Yung, "Scalable group signatures with almost-for-free revocation," in *Annual Cryptology Conference*, pp.571-589, 2012.

[30] T. Nakanishi, N. Funabiki, "Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 533-548, 2009.

[31] T. Nakanishi, H. Fujii, Y. Hira, N. Funabiki, "Revocable group signature schemes with constant costs for signing and verifying," in *International Workshop on Public Key Cryptography*, pp. 463-480, 2009.

[32] L. Nguyen, "Accumulators from bilinear pairings and applications," in *Cryptographers Track at the RSA Conference*, pp. 275-292, 2005.

[33] A. Shamir, Y. Tauman, "Improved online/offline signature scheme," in *Annual International Cryptology Conference*, pp. 355-367, 2001.

[34] F. Zhang, K. Kim, "ID-based blind signature and ring signature from pairings," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 533-547, 2002.

[35] S. Zhou, D. Lin, "Shorter verifier-local revocation group signatures from bilinear maps," in *International Conference on Cryptology and Network Security*, pp. 126-143, 2006.

Biography

Ke Gu received his Ph.D. degree in School of Information Science and Engineering from Central South University in 2012. He is currently an associate professor at Changsha University of Science and Technology. His research interests include cryptography, network and information security.

Dianxing Liu is currently an associate professor at Hunan Police Academy. His research interests include network analysis.

Bo Yin is currently an assistant professor at Changsha University of Science and Technology. His research interests include data analysis, network and information security.

Internet of Things: A Secure Cloud-based MANET Mobility Model

Tanweer Alam
Department of Computer Science & Islamic University of Madinah
170, Prince Naif Bin Abdulaziz Road, Madinah, Saudi Arabia
(Email: tanweer03@iu.edu.sa)

(Received Mar. 31, 2018; Revised and Accepted Feb. 9, 2020; First Online Feb. 28, 2020)

Abstract

Connected devices such as smart home automation gateway, smart air conditioners, smart hubs, smart thermostat, color changing smart lights, smart mobile phones, smart watches and smart tablets, *etc.* are omnipresent in our everyday lives and are becoming a valuable tool with wireless networking features using different wireless protocols commonly used. Access points allow interactions between users within an Internet of Things ecosystem infrastructure. These smart devices are automatically connected, and a network is formed by themselves. However, there are many challenges throughout this established network of its own for secure communication. Security has been perceived as a popular barrier to adopting the cloud model of internet realism. The storage and resource management may be in the cloud environment is a distributed structure that places the world in a raised situation with many concerns over its weaknesses, security risks and difficulties. Different participation parties have broadened those issues depending on the viewpoint and goal of each party. The author primarily addresses the causes of challenges and difficulties related to security, reliability, privacy and availability of services from the Cloud point of view. Connectivity Security has been identified as one of cloud computing's most critical issues where resolving such an issue would result in constant growth in the use and popularity of cloud computation. The purpose of this study is to build a mobile ad hoc network mobility model framework using cloud computing to provide secure Internet of Things communication between smart devices. The major contribution relates a new methodology to ensure secure communication with the 5G network of smart devices using the internet. The approach uses a desired study's accurate and effective simulation and can be applied in an Internet of Things structure. This research would create a new connectivity architecture to address the problem of secure communication between smart devices in 5G networking.

Keywords: Cloud Computing; Internet of Things; Mobile Ad-Hoc Networks (MANET); Mobility Models; Smart Devices; 5G Heterogeneous Network

1 Introduction

This research is a move forward into the field of cloud computing and the Internet of Things in 5G heterogeneous networks as the author suggested a framework for the mobility model using cloud computing to connect smart devices together on the internet. In this study, the proposed research work is an improvement and implementation of current mobile ad-hoc network communication using the cloud within the internet of things environment. This research provides an approach that is able to develop a new framework for the secure communication of smart devices on the internet. This research was used in the right and efficient simulation of the targeted research and can be applied within the IoT framework. Today's wireless network is composed of cells in a specific area within its range. Every cell includes a base station, which can be linked through wired or wireless networks [29]. Nowadays, smart devices provide very useful Wi-Fi Direct functionality [1]. By using this technology any device can communicate and create a MANET network with neighborhood devices [9, 17]. When one device has internet then the same device can connect to the cloud and build a MANET of the smart device [20]. The growth of the internet of smart devices is estimated to increasingly connect with 50 billion smart devices by 2020 [8]. Such growth would not rely on the population of humankind, however on the fact that we regularly use smart devices [5].

A reality of objects that are connected is that they can communicate between the device to the device [4]. These devices are going to talk to one another [3]. However, one of the most comprehensive issues is the monitoring and tracking of movable devices [16]. A concept of the internet of things could be represented as "a ubiquitous and pervasive network that further facilitates the monitoring of physical devices and control through gathering, processing, and also analyzing the information using sensor network" [12, 13]. The evolutionary paradigm allows the users to deliver effortless access to a network of computing resources, where users can easily scale up or down their expectations with the service provider's irrelevant interactions [15].

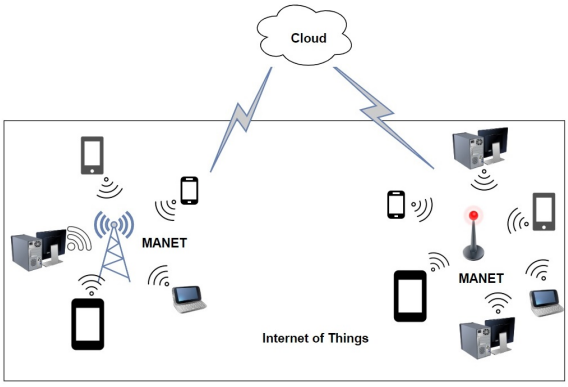


Figure 1: Internet of things with cloud and MANET

Started from 2008, the internet of things is growing exponentially to connect the physical objects using the internet. These physical objects are connected to an intelligent database that has smart data storage [7]. To recognize the physical object, houses, people's image, location *etc.* the framework uses image recognition techniques. The internet of things is now moving from information technology to operational technology, *i.e.* IPV4 (man 2 machines) to IPV6 (machines 2 machines) and It integrates sensors, smart devices and such Smart Grid interfaces [24]. Throughout a general sense, each user has questions about flaws in cloud computing and challenges that might keep them from achieving their goals. The three innovations that have contributed to growth on the internet of things. A ubiquitous computation that has the intelligent capability of physical objects operating on the computation platform [19]. The Internet Protocol (IPV6) uses universal computing covering the network area and enabling machine-to-machine communication. The IPv4 internet has a weakness of connecting billions of smart devices together, but IPv6 internet is possible because it enables the internet of things to link billions of smart devices together [22]. Communication in ubiquitous computing with the use of sensor connectivity in fixed cell networks or mobility should be periodically improved to allow the advancement of smart devices internet including multi-sensor systems to store, compute, analyze and process capabilities that require smaller size and lowest energy [23]. This article shows the main contribution that links a new secure model of communication using cloud computing and MANET technologies in the Internet of Things environment [34]. A secured data self-destructing scheme in Cloud computing is presented in the article [31].

The idea of communication security depends on three main points in the architectural design of the internet of things.

1) Managing information receiving from millions of sensors in a centralized smart device collection system is not easy.

2) Managing network resources in a large network that can gather environment information from the centralized framework is not an easy task.

3) Managing sensors which execute the same kind of data-parallel and stored in the centralized system is very complicated [26].

Cloud computing would have been one of the most famous concepts in computation [25]. This also comes out of recent computing paradigms advances that combine parallel computing, grid computing, disseminated computing, and other computing methodologies. The cloud computing provides its users with three basic models of administration: SaaS, PaaS, and IaaS. The first one is SaaS, Software as a service (SaaS) is intended primarily for end-users who need to use the software as part of their everyday lives. Secondly, Platform as a service (PaaS) is primarily intended for software developers who need technologies to develop their software as well as implementation. The principal aim of Infrastructure as a Service (IaaS) is to network architects who need infrastructure functionality [10]. A more essential element of the framework is the communication security challenges and threats to communicate smart device's internet from a cloud point of view. The cloud environment offers the power of shared resources to its end users. The cloud providers employ multi-tenancy to eventuate the concept of exchanging. Basically, the implementation of a maximum throughput of recourses is provided as software architecture.

MANET is a very successful system to always get connected at any time anywhere. The cloud computing provides data storage and access service. Cloud and MANET integration provides the services for cloud access within MANET of devices connected. The smart devices are able to connect to each other in the area where there are no network facilities [11]. Among smart devices, the MANET can be formed automatically. And it can use the cloud service if one device has the internet in the group of smart devices. MANET mobility model implementation in the smart device to smart device communication can be very efficient and useful to save energy as well as increase the efficiency in the Internet of Things. This approach of cloud-based services in the MANET model for the device to device communication can be a very useful approach to enhance the capabilities of smart devices in the internet of things environment. MANET connected devices are also able to use cloud service to discover neighborhood devices and exchange information. The proposed approach includes MANET and cloud computing on the internet of smart devices that can be useful in the 5G heterogeneous network.

The rest of the paper is organized as follows: Section 2 shows the literature survey, Section 3 presents the methodologies, Section 4 presents the Cloud-MANET Mobility Model and Section 5 represents the conclusion of the research.

2 Literature Survey

In 2012, Lacuesta *et al.* was published an article [30] on the internet of things trust. The smart devices that can take part in the MANET networks can be quite different from one another. The Network can be embedded with sensors, mobile apps, home appliances, or other types of devices, will need to work together to increase and enhance customer satisfaction. Some of these tools may have limited resources to run, sometimes even non-existent resource constraints, they must work to optimize network traffic. The authors were focusing attention on spontaneous networks in this article. They were proposed a secure ad hoc spontaneous network, based on direct peer-to-peer interaction in the Internet of Things [30]. A secured and authenticated anonymous data access on cloud in the MANET network is presented in the article [28]. In the article [32], the zone Based MANET Routing Protocol with a Genetic algorithm is presented.

In 2015, D. Airehrour, *et al.* was published an article [2], they were developed various secure routing protocols for MANETs which could be used to establish secure routing protocols for the Internet of Things, so the analysis of these secure MANET routing protocols will provide a roadmap for the development and implementation of security in the Internet of Things and Cloud Computing. In this paper, the authors also provide secure routing protocols in MANETs while offering some secure routing features for IoT routing to ensure confidentiality and integrity. They also discussed research trends and future directions in the field of IoT network security [2].

In 2017, R. Al-Zaidi, *et al.* was published an article [27], the authors were presented the Internet of Things (IoT) technology over Ship Ad-hoc Networks (SANET) as a maritime data acquisition and cartography system The Ships were recommended to communicate over a Very High Frequency (VHF) which is already available on most ships and are fitted with multiple sensors such as sea level, temperature, wind speed and direction and so on. 5G base station nodes onshore server sinks for the data collected and are fitted with data aggregation and processing capabilities for Mobile Edge Computing. Finally, the sensory data is aggregated on the internet in a central repository to generate up-to-date digital cartography solutions.

In 2018, Tanweer Alam and Mohamed Benaida have published an article [18], they were proposed a framework that can access and deliver cloud services to the MANET users through their smart devices in the IoT. Also, the proposed framework was performed where all computations, data handling, and resource management. MANETs can connect to the cloud and can use cloud services. The main contribution in this research links a new methodology for providing secure communication on the internet of smart devices using MANET. In this research, the methodology uses the correct and efficient simulation of the desired study and can be implemented in a framework of the Internet of Things in the future.

In 2020, H. Riasudheen, *et al.* are published an article [33], the Device-to-Device (D2D) interaction in 5G networks has increased the number of devices and the rate of data transmission in Cloud Assisted-Mobile Ad Hoc Networks of smart devices. Furthermore, due to mobility, connection failure overload networking and limited battery energy, the connection between the smart devices has to be regularly renewed. It consumes lots of energy during this time in searching for and connecting the smart devices. Compared with other existing network mobility models and routing protocols, this proposed research provides better performance.

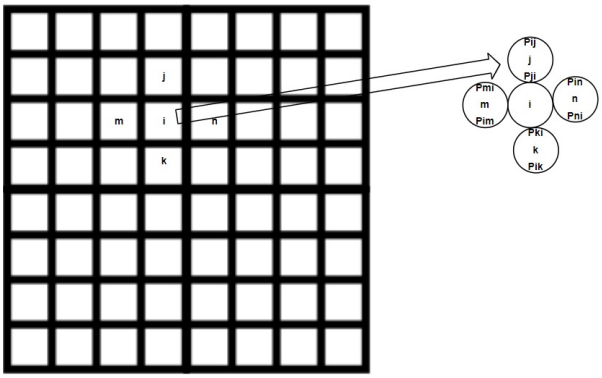


Figure 2: HMM model transition matrix representation

3 Methodologies

Discovering the smart devices in MANET, the hidden Markov model is utilized in the 2Dimensional plane zone. The framework is connected in this area and smart devices can move inside this area and search another smart devices. The transition matrix is formed in the area of MANET for discovering all the smart devices. Some parameters are used for discovering smart devices as follows.

Suppose, S=$S_1, S_2, \cdots, S_N$ where S is the state, S_1 is the first state, S_2 is the second state, *etc.* Every cell depends on one state in S.

Transition matrix probability P=$P_{ij}(1 <= i <= N)$ where P_{ij} characterized to move likelihood from S_i to S_j in the Transaction Matrix.

The probability P_{ij} is just significant if S_i, S_j is neighborhood states in the 2D plane. The states will rearrange to move up, down, left and right in the 2D plane [14]. The left components within the framework are all 0s initially. The hidden Markov model is represented in the following transition matrix.

The smart device in each cell represented by $\pi = \pi_i(1 <= i <= N)$.

Smart devices in MANET can be used to discover the signals using the Viterbi algorithm in the 2D plane. Suppose $O_1, O_2, ..., O_n$ are the observation of discovering the devices in the 2D plane. Each smart device sends a report of observations during a time period [6]. The algorithm discovers the way at each step by maximizing the throughput of the smart devices. The process is too much time

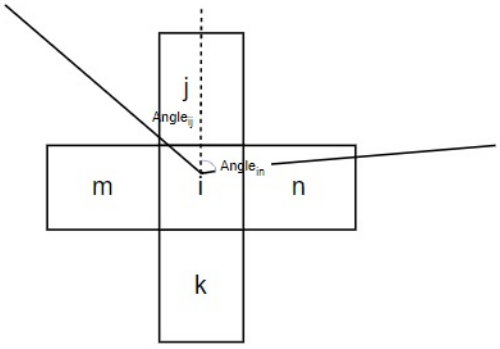


Figure 3: Motion in gradient-based model

consuming for the rush of devices in the specific zone. This whole process can be accomplished by essentially joining the sub-track at every cell of the area. The fact is that the HMM model has been utilized as a part of the target to discover the device connected to the appropriate MANET of smart devices [21]. Hidden Markov model depends on the state's probability and the transition matrix represents the information in every cell of the area. If a smart device enters a new cell then it removes previous data and updates the information.

Discovering the smart devices, the gradient model works to find the devices and share the information among the smart devices. Eventually, when a smart device recognizes another smart device, the gradient value will set 1 also discover another smart device in the area where MANET is formed.

Apply the physical law and find the distance between the two smart devices that is proportional to one upon the distance of the event.

Device (distance) $\propto$ 1/eventdistance.

Gradient over time is represented by the following formula.

The gradient with respect to time t as follows:

$$Gradient(t) = \left\{ \begin{array}{l} 1 \text{ if } t = 0 \\ event - t \text{ if } 0 < t < totaltime \\ 0 \text{ if } t >= totaltime \end{array} \right\} \quad (1)$$

The Gradient model finds the gradient distribution over the time period. When t=0 then the gradient value will be 1 and if time is greater than the total time then the value of gradient will be 0, otherwise, the gradient is proportional to one upon time power of the event (event-t).

Smart devices are held in the range of MANET that considers coverage and connectivity of the Wi-Fi ad hoc network. Each smart device is expected to have a settled Wi-Fi area. Firstly, the wi-fi Ad Hoc Network is dynamically connected. The MANET is inactive all the time, when one device wants to make the connection with another device then it creates a connection with their neighborhood device. The inactive smart device additionally occasionally awakens to enter into an active device. The active state detects the devices and connects dynamically.

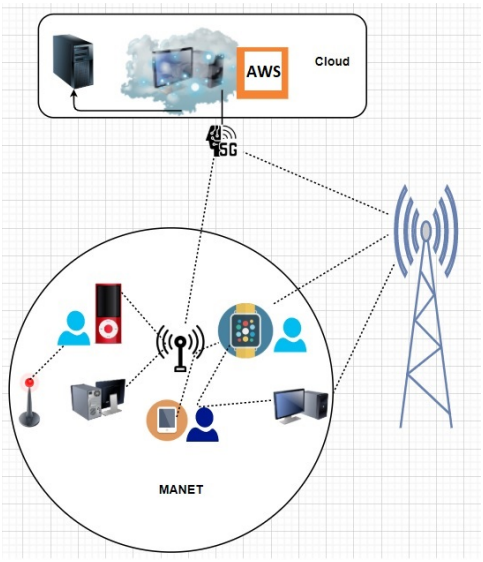


Figure 4: Cloud-MANET mobility model

4 Cloud-MANET Mobility Model

Device-to-device connectivity will improve the overall performance, expand coverage and minimize the energy consumption of mobile communications through communicating directly. Smart device to smart device communication in the cloud-MANET framework of the internet of things is a novel methodology that discovers and connected nearby smart devices without a centralized system. In the existing cellular network, not allowed to connect all smart devices without centralized infrastructure even if they are very near. This proposed technique would be very useful in the machine to machine (M2M) communication because, in the M2M network, there are many devices nearby. Therefore, the implementation of the MANET model in the smart device to smart device communication can be very efficient and useful to save power as well as efficiency. Cloud-based MANET mobility model for the device to device communication can be a very useful approach to enhance the capabilities of smart devices in 5G.

When two or more smart devices wish to communicate in the Cloud-MANET model on the internet of things then communication security is the main challenge. Throughout the Cloud-MANET model, the smart devices are dynamically joined and created a network on their own. Also, they can access the cloud service. But there are more challenges for secure communication in this own created network that access cloud services. Cloud-MANET is a kind of wireless network that is self-organizing and auto connected in a decentralized system and accessed the cloud. Each device in MANET can be moved freely from one location to another in any direc-

tion within the range of Wi-Fi. Several MANETs can connect with the same cloud and they can use cloud services. MANET model of smart devices in local communication can work very well using the cloud, it is failed when it connects in exist wired networks. Every smart device needs to search for neighborhood devices. A most important question arises here: How will communication secure in the public cloud and MANET? The answer is yes, it is possible through the cloud-MANET model that is implemented and integrated with mobile apps and tested. Cloud MANET mobility model is an integrated model of Cloud computing and MANET networks.

MANET model is depended on the mobility of its nodes and connectivity, resources such as storage and power consumption. Cloud providers retain network infrastructure, storage facilities, and software applications that support flexibility, efficiency, and ability. The Cloud-MANET mobility model allows the smart devices to communicate with another smart device, However, at least one smart device must be connected to cellular or Wi-Fi networks and access the internet. Every smart device of MANET should be registered in the cloud. If a MANET is activated then cloud services will activate in real-time and provide services to the smart devices of MANET to communicate. Smart devices send a request to the cloud to provide a session of connection. The connection can be described as the probabilistic function as follows.

$$\begin{aligned} & Session(life)) \\ = & \left\{ \left(\frac{\int_{life}^{\infty} \left(\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) erf\left(\log\left(\frac{u}{\mu}\right) \div \sqrt{2}\sigma\right)\right) du}{\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) erf\left(\log\left(\frac{life}{\mu}\right) \div \sqrt{2}\sigma\right)} \right) \right\} \end{aligned} \quad (2)$$

The integral limit is 0 tends to ∞ in the session life. The above probabilistic method requires to compute the values of σ and μ for every smart device when they have a session with the cloud.

σ and μ are two parameters that are related to the connection establishment among MANETs and Cloud service. It can be measured through smart devices using the following method.

$$e^{\mu + (1/2)\sigma^2}$$

If a smart device estimates the connection life between MANET and Cloud then it will transfer or receive secured data.

Connection will be active and can connect. The author is considered that every smart device is assured to establish the route between MANET and cloud when they create a session. Smart devices can move through the maximum speed from one location to another location by using the Gauss-Markov mobility framework. Calculate the moving speed and direction of the smart device within the MANET range by using the following formula.

$$\begin{aligned} & Speed_t \\ = & \lambda Speed_{t-1} + (1-\lambda)\, \bar{Speed} + \sqrt{(1-\lambda^2)\, Speed_{t-1}^{G}} \end{aligned}$$

and

$$\begin{aligned} & Direction_t \\ = & \lambda Direction_{t-1} + (1-\lambda)\, \bar{Direction} \\ & + \sqrt{(1-\lambda^2)\, Direction_{t-1}^{G}} \end{aligned}$$

λ is used as a random degree when computing speed as well as the direction of smart device in time (t).

Transmission (t_s) of information (I_k) among the number of smart devices (S_n) can be estimated during the time interval $[t_i, t_{i-1}]$.

Smart devices can move within the MANET and access the cloud service using the multidimensional function (ε_k).

$$\varepsilon_k = \mathbb{C}^{S_n \times t_k} \times I_k.$$

Where k=0,1,2,3.....∞ (+ve).

When smart devices have moved outside the MANET then k will be negative. We have considered that the transformation of information simultaneously happens. The probability is proportional to the one divide by information (I_k).

Discover the smart device and find the new position of the smart devices in MANET using the following algorithm (Algorithm 1).

This Cloud-MANET mobility model had been implemented and tested. The MANET is building and verified on three Samsung mobile phones. The Amazon Web Services (AWS) are used for implementing cloud services with MANET. The amazon cloud service will connect to the MANETs and provide a session for connection. At least one device should be registered in the cloud and access the cloud and share it in MANET.

5 Conclusions

Cloud-MANET mobility model can play a vital role in 5G. It can enhance the efficiency and speed of communication in the cloud and MANET. The cloud paradigm is based on a distributed architecture, it is inherited some risks and vulnerabilities that are related to distributed computing. Communication security threats and challenges that rely on behind the lure of cloud computation. Cloud-MANET mobility model has been developed and tested. One device start service of MANET as well as connected with the cloud and start to share connection and exchange information. The author has used cloud service from Amazon cloud. This study showed successfully and expectation for a future scope in this field. The author reached the conclusion that this kind of network could help people in many situations, some of them in critical situations after researching a lot of how MANET networks work and which are its advantages and disadvantages.

Algorithm 1 Discover the smart device

Begin
Find the position (X_1, Y_1) of the smart device in the MANET.
Find the current Speed (s) of the moving device in MANET.
Speed can get using the following formula.
Speed (s)= distance (d)/time (t).
while time=t and angle θ is positive **do**
  we consider the new location of the smart device using the following formula.
  $X_2 = X_1 + s * t * cos(\theta)$
  $Y_2 = Y_1 + s * t * sin(\theta)$;
  if θ is negative **then**
    $X_2 = X_1 - s * t * cos(\theta)$
    $Y_2 = Y_1 - s * t * sin(\theta)$;
  end if
end while
Obtain the real Location of smart device on x, y axis;
Obtain the theoretical location on x, y axis;
Obtain the distance d between Loc and ref.
Distance (d)=$\sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}$
Obtain the random location (X, Y) of smart device at the diagonal of triangle can find using the following formula.
$X = Math.random(d.getX())$;
$Y = Math.random(d.getY())$;
Obtain the actual location of the smart device according to the diagonal of the triangle, it may be on the diagonal or upper or lower than the diagonal.
if the device is upper **then**
  The diagonal then increase the value of X and Y
  X= X + δX;
  Y= Y + δY ;
else
  X= X - δX;
  Y= Y - δY;
end if
Return new Location(X, Y).
End

Acknowledgments

The author gratefully acknowledges the anonymous reviewers for their valuable comments.

References

[1] O. Abu-Sharkh, E. Qaralleh, and O. Hasan, "Adaptive device-to-device communication using Wi-Fi Direct in smart cities," *Wireless Networks*, vol. 23, no. 7, pp. 2197-2213, 2017.

[2] D. Airehrour, and J. Gutierrez, "An analysis of secure MANET routing features to maintain confidentiality and integrity in IoT routing," in *International Conference on Information Resources Management (Conf-IRM'15)*, 2015. (https://pdfs.semanticscholar.org/4fce/d5fcdeab6334cfbef523634765b786c740ac.pdf)

[3] T. Alam, "A reliable communication framework and its use in internet of things (IoT)," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT'18)*, vol. 3, no. 5, pp. 450-456, 2018.

[4] T. Alam, "A reliable framework for communication in internet of smart devices using IEEE 802.15.4," *ARPN Journal of Engineering and Applied Sciences*, vol. 13, no. 10, pp. 3378-3387, 2018.

[5] T. Alam, "Blockchain and its role in the internet of things (IoT)," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 5, no. 1, pp. 151-157, 2019.

[6] T. Alam, "Cloud computing and its role in the information technology," *IAIC Transactions on Sustainable Digital Innovation (ITSDI'20)*, vol. 1, no. 2, pp. 108-115, Feb. 2020.

[7] T. Alam, "Fuzzy control based mobility framework for evaluating mobility models in MANET of smart devices," *ARPN Journal of Engineering and Applied Sciences*, vol. 12, no. 15, pp. 4526-4538, 2017.

[8] T. Alam, "IoT-Fog: A communication framework using blockchain in the internet of things," *International Journal of Recent Technology and Engineering (IJRTE'19)*, vol. 7, no. 6, 2019.

[9] T. Alam, "Middleware implementation in cloud-MANET mobility model for internet of smart devices," *International Journal of Computer Science and Network Security*, vol. 17, no. 5, pp. 86-94, 2017.

[10] T. Alam, "Tactile internet and its contribution in the development of smart cities," *Computer Science*, 2019. (https://www.researchgate.net/publication/333915765_Tactile_Internet_and_its_Contribution_in_the_Development_of_Smart_Cities)

[11] T. Alam, "5G-Enabled tactile internet for smart cities: Vision, recent developments, and challenges," *Journal Informatika*, vol. 13, no. 2, pp. 1-10, 2019.

[12] T. Alam and M. Aljohani, "An algorithm for accessing traffic database using wireless technologies," in *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC'15)*, pp. 1-4, 2015.

[13] T. Alam and M. Aljohani, "An approach to secure communication in mobile ad-hoc networks of Android devices," in *International Conference on Intelligent Informatics and Biomedical Sciences (ICI-IBMS'15)*, pp. 371-375, 2015.

[14] T. Alam and M. Aljohani, "Decision support system for real-time people counting in a crowded environment," *International Journal of Electronics and Information Engineering*, vol. 12, no. 1, pp. 34-41, Mar. 2020.

[15] T. Alam and M. Aljohani, "Design a new middleware for communication in ad hoc network of android

smart devices," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, pp. 38, 2016.

[16] T. Alam and M. Aljohani, "Design and implementation of an ad hoc network among Android smart devices," in *International Conference on Green Computing and Internet of Things (ICGCIoT'15)*, pp. 1322-1327, 2015.

[17] T. Alam, M. Benaida, "CICS: Cloud–internet communication security framework for the internet of smart devices," *International Journal of Interactive Mobile Technologies (iJIM'18)*, vol. 12, no. 6, pp. 74-84, 2018.

[18] T. Alam, and M. Benaida, "The role of cloud-MANET framework in the internet of things (IoT)," *International Journal of Online and Biomedical Engineering (iJOE'19)*, vol. 14, no. 12, pp. 97-111, 2019.

[19] T. Alam, P. Kumar, and P. Singh, "Searching mobile nodes using modified column mobility model," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 1, pp. 513-518, 2014.

[20] T. Alam, B. Rababah, "Convergence of MANET in communication among smart devices in IoT," *International Journal of Wireless and Microwave Technologies (IJWMT'19)*, vol. 9, no. 2, pp. 1-10, 2019.

[21] T. Alam, A. A. Salem, A. O. Alsharif, and A. M. Alhejaili, "Smart home automation towards the development of smart cities," *APTIKOM Journal on Computer Science and Information Technologies*, vol. 5, no. 1, 2020.

[22] T. Alam, and B. K. Sharma, "A new optimistic mobility model for mobile ad hoc networks," *International Journal of Computer Applications*, vol. 8, no. 3, pp. 1-4, 2010.

[23] T. Alam, P. Singh and P. Kumar, "Generating different mobility scenarios in ad hoc networks," *International Journal of Electronics Communication and Computer Technology*, vol. 4, no. 2, 2014.

[24] T. Alam, A. P. Srivastava, S. Gupta, and R. Tiwari, "Scanning the node using modified column mobility model," *Computer Vision and Information Technology Advances and Applications*, 2007. (`https://www.researchgate.net/publication/306440139_Scanning_the_Node_Using_Modified_Column_Mobility_Model`)

[25] M. Aljohani and T. Alam, "Design an M-learning framework for smart learning in ad hoc network of Android devices," in *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC'15)*, pp. 1-5, 2015.

[26] M. Aljohani, and T. Alam, "Real time face detection in ad hoc network of Android smart devices," in *Advances in Computational Intelligence: Proceedings of International Conference on Computational Intelligence*, pp. 245-255, 2016.

[27] R. Al-Zaidi, J. Woods, M. Al-Khalidi, K. M. A. Alheeti and K. McDonald-Maier, "Next generation marine data networks in an IoT environment," in *The Second International Conference on Fog and Mobile Edge Computing (FMEC'17)*, pp. 50-55, 2017.

[28] M. P. D. Bardiya, and P. Ramteke, "Secured and authenticated anonymous data access on cloud in MANET," *International Journal of Advanced Research in Computer and Communication Engineering (ISSN'15)*, vol. 2278, no. 1021, pp. 128-137, 2015.

[29] E. B. Brownrigg and T. W. Wilson, *Wireless Network System and Method for Providing Same*, US6044062A, 2000.

[30] R. Lacuesta, G. Palacios-Navarro, C. Cetina, L. Peñalver and J. Lloret, "Internet of things: Where to be is to trust," *EURASIP Journal on Wireless Communications and Networking*, no. 203, 2012.

[31] L. Liu, Y. Li, Z. Cao and Z. Chen, "A note on one secure data self-destructing scheme in cloud computing," *International Journal of Network Security*, vol. 22, no. 1, pp. 36-40, 2020.

[32] V. Preetha, and K. Chitra, "ZBMRP: Zone based MANET routing protocol with genetic algorithm and security enhancement using neural network learning," *International Journal Network Security*, vol. 20, no. 6, pp. 1115-1124, 2018.

[33] H. Riasudheen, K. Selvamani, S. Mukherjee and I. R. Divyasree, "An efficient energy-aware routing scheme for cloud-assisted MANETs in 5G," *Ad Hoc Networks*, vol. 97, pp. 102021, 2019.

[34] A. Sharma, T. Alam, and D. Srivastava, "Ad hoc network architecture based on mobile Ipv6 development," *Advances in Computer Vision and Information Technology*, 2007. (`https://www.researchgate.net/publication/306440273_Ad_hoc_Network_Architecture_Based_On_Mobile_IPv6_Development`)

Biography

Tanweer Alam is with the Department of computer science, Islamic University of Madinah since 2013. He is awarded by Ph.D. (Computer Science and Engineering), M.Phil. (Computer Science), MTech (Information Technology), MCA (Computer Applications) and M.Sc. (mathematics). His area of research including Mobile Ad Hoc Network (MANET), Smart Objects, Internet of Things, Cloud Computing and wireless networking. He is a single author of twelve books. He is the member of various associations such as International Association of Computer Science and Information Technology (IACSIT), International Association of Engineers, Internet Society (ISOC), *etc.*

Reversible Data Hiding Schemes in Encrypted Images Based on the Paillier Cryptosystem

Hefeng Chen[1], Chin-Chen Chang[2], and Kaimeng Chen[1]
(Corresponding author: Chin-Chen Chang)
Computer Engineering College, Jimei University[1]
Xiamen 361021, PR China
Department of Information Engineering and Computer Science, Feng Chia University[2]
Taichung, Taiwan
(Email: alan3c@gmail.com)
(Received Aug. 19, 2018; Revised and Accepted Aug. 10, 2019; First Online Feb. 28, 2020)

Abstract

In this paper, we propose a novel framework for reversible data hiding schemes in encrypted images inspired by the privacy needs of outsourcing data in the cloud service. Our scheme allows the image owner and the data provider to send encrypted images and encrypted data to the data processor separately; then, the data processor can do the embedding without knowing any side information; the receiver would obtain the marked image after decryption and could extract the hidden data and completely recover the original image. By exploiting the Paillier homomorphism and the equivalence of the modular approach, the high capacity of at least 1 bpp and even exceeding 1016 bpp can be achieved at one-time embedding. Then, we extended the first scheme to provide a multi-receiver, reversible data hiding scheme by combining our approach with the(t, w)-threshold secret sharing homomorphism. It is suitable for the application of distributed storage with fault tolerance or the protection of patients' privacy when they are consulting with multiple doctors.

Keywords: Homomorphic Encryption; Reversible Data Hiding; Secret Sharing

1 Introduction

Reversible data hiding (RDH) is a technique that embeds secret data into the cover medium in a reversible manner. In the RDH scheme, the embedded data can be extracted correctly, and, also, the cover medium can be recovered perfectly from the marked data. Prior studies have proposed several approaches for RDH, such as difference expansion [7, 11], lossless compression [21], histogram shifting [15], and prediction error expansion [4]. Motivated by the need to preserve privacy in cloud computing and other applications for securely storing or sharing multimedia files with others, the combination of data hiding and encryption has received increasing attention. RDH for encrypted images enables cloud servers to reversibly embed data into images, but no knowledge about image content is available.

The first encrypted image-based RDH scheme was proposed by Puech *et al.* [20], who used the bit substitution method to embed one bit into a block of pixels encrypted by Advanced Encryption Standard. The extraction process is just simple read, and the decryption process is done by analyzing the local standard deviation. In Zhang's scheme [30], the bits of each pixel are encrypted by exclusive-or with pseudo-random bits, and then, the encrypted image is partitioned into blocks. An additional bit is embedded into each block one by one by flipping a portion of the least significant bits (LSBs). The extraction and decryption can be done by examining the fluctuation in natural image blocks. Then, the higher embedding capacity with a lower bit error rate is achieved by defining different evaluation functions based on the spatial correlation of blocks [22], by using a different flipping strategy [12], or by using prediction error [28].

Qin and Zhang [22] proposed the flipped pixels' elaborate selection method to improve the visual effect of the decrypted. Zhou *et al.* [32] proposed a scheme with a high embedding capacity by utilizing a public-key modulation mechanism without sharing the secret data hiding key and a two-class SVM classifier for decoding. In addition, Ma *et al.* [16] reserved room before encryption to obtain large payloads up to 0.5 bit per pixel, and the performance was improved further by considering patch-level sparse representation [6].

However, all of the images are encrypted with symmetric cryptosystem in [1, 6, 12, 16, 20, 22, 28, 30, 32], making it difficult for them to be processed directly in the encryption domain. This disadvantage can be overcome by introducing the homomorphic encryption. In order to process the encrypted data directly, the special functions called "privacy homomorphism" [23] must be found. In other

words, after the ciphertext is processed, an encrypted result is generated that matches the desired plain-text result after decryption. Since the encrypted image can be processed directly, the privacy and confidentiality of the user can be enhanced. Hence, conducting RDH in the homomorphic encryption domain can enrich its availability in cloud computing and other similar scenarios.

Recently, an additive homomorphic Paillier cryptosystem-based RDH scheme [19] also has been investigated [9, 14, 27, 31]. First, Chen *et al.* [9] designed the RDH with the public-key cryptosystem by dividing each pixel value into two portions, *i.e.*, the seven most significant bits (MSBs) and one LSB, and then they performed the encryption using the Paillier cryptosystem. Then, two encrypted LSBs of each encrypted pixel pair are modified to reversibly embed one additional bit following the homomorphism.

Zhang *et al.* [31] used histogram shrink before encryption and used error-correction codes to expand the additional data to achieve reversibility. Wu *et al.* [27] presented two high-capacity RDH schemes, one by doing value expansion on the encrypted pixel values and another by taking advantage of the self-blinding feature of the Paillier encryption. Both embedding capacities are more than 1 bpp. Li *et al.* [14] used histogram shifting in encrypted images to embed bits. Compared with the image RDH algorithms with symmetric cryptography, the proposed algorithms are more suitable for the cloud environment without reducing the security level.

The interpolation-based RDH is also an important work [1, 10, 13, 24]. This paper mainly uses polynomial interpolation technique to realize secret sharing and then solve the RDH problem for multiple receivers.

The (t, w) secret sharing scheme was developed by Shamir [25] based on polynomial interpolation, and it was developed independently by Blakley [5] in 1979 based on geometry. The basic idea is to protect the privacy of information by distribution. In a (t, w) secret sharing scheme, a dealer divides a secret into w shares and the secret is shared among a set of w shareholders, in such a way that any t or greater shareholders can reconstruct the secret, while fewer than t shareholders cannot.

There are other types of secret sharing, *e.g.*, McEliece-Sarwate's scheme [17], which is based on Reed-Solomon codes, and Mignotte's scheme [18] and Asmuth-Bloom's scheme [2], which are based on the Chinese remainder theorem (CRT). In 1987, Benaloh [3] first proposed the concept of secret sharing homomorphism, which allows multiple secrets to be combined by direct computation of shares. This property reduces the need for trust among the agents. Some secret sharing-based, data-hiding algorithms have been presented in literature [8, 26]. Recently, Wu *et al.* [29] introduced a model of RDH in encryption domain-based secret sharing. The image content owner encrypts the original image into several shares and sends them to the service provider. The servive provider is responsible for storing and reversibly hiding data into encrypted shares, extracting the hidden data, and sending the encrypted shares to the authenticated receiver who can recover the desired image. This model can be applied to the scenario in which extraction is required for image decryption.

RDH in the encrypted domain is suitable for the scenario in which the image owner and the data hider are not the same person. The image owner would encrypt the medium before transmission, and the data hider can append some additional message into the cipher without knowing the plaintext image. Then, the receiver can recover the original image and extract the embedded data losslessly.

In this paper, we address the issue concerning the separation of the roles of the data provider and the data processor, and both images and data are encrypted before transmission to the data processor. The data processor does not know anything about the image or the hidden data but can integrate them to a new cipher in a way that the receiver can perfectly decrypt the image and extract the data. For example, in electronic-healthcare, medical images and electronic patient records are generated by two different departments, and the information should be encrypted before it is transmitted to the database administrator to protect the patient's privacy.

The database administrator embeds the patient's encrypted record into the corresponding encrypted image to achieve privacy homomorphism. Then, when the doctor receives the marked encrypted medical image, he or she can get the original medical image and data. Our scheme is suitable for the scenario in which image decryption is required for extraction. Also, considering the application scenario after consultation with several doctors, we propose a (t, w) multi-receiver RDH scheme using secret sharing homomorphism to achieve the goal that any t receivers can collaborate with each other by using their shadows to reconstruct the original image and extract the hidden data, which cannot be done unless t or more receivers cooperate.

In short, there are two contributions of our work:

1) Proprose a RDH scheme suitable for data outsourcing, in which the roles of data owner and data hider are separated.

2) Extend the RDH scheme for multi-recever case, combined with the secret sharing technology.

The rest of this paper is organized as follows. Section 2 gives some preliminaries. In Section 3, we review the related works proposed by Chen *et al.* [27] and by Li *et al.* [14]. In Section 4, we propose a high-capacity RDH scheme based on the Paillier cryptosystem. In Section 5, we present another scheme for sharing the marked encrypted image among multiple receivers who have the same decryption key. The performance analysis and the experimental results are shown in Section 6, and our conclusions are made in Section 7.

Table 1: Notations

N	RSA modulus, $N = p \cdot q$, where p and q are two large primes, while $(p-1)/2$ and $(q-1)/2$ are also primes
$\mathbb{Z}_N$	Integers modulo N, $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$
$\mathbb{Z}_N^*$	Multiplicative group of $\mathbb{Z}_N$, $\mathbb{Z}_N^* = \{r \in \mathbb{Z}_N \mid gcd(r, N) = 1\}$
$\varphi(\cdot)$	Euler's phi function, $\varphi(N) = (p-1)(q-1)$
$\lambda(\cdot)$	Carmichael's function, $\lambda(N) = lcm(\varphi(p), \varphi(q))$
$L(\cdot)$	$L(u) = (u-1)/N, \forall u \in \{u < N^2 \mid u = 1 \bmod N\}$
e_{PK}	Encryption algorithm with the receiver's public key PK
d_{SK}	Decryption algorithm with the receiver's private key SK
$\lfloor \cdot \rfloor$	Floor function

2 Preliminaries

Two important techniques were used to design our proposed scheme, *i.e.*, homomorphic cryptosystem and the secret sharing scheme. The former allows direct processing in the encryption domain to reach privacy homomorphism [23], that is, an encrypted result will generate a decryption that matches the desired result without knowing the decryption key. Although Goldwasser-Micali scheme is the classical homomorphic encryption, it only supports additive homomorphism on $\mathbb{Z}_2$ domain, so we finally choose Paillier system. The latter can decompose one secret into shadows that are distributed among shareholders, such that the pooled shadows of specific subsets of users allow the reconstruction of the original secret. To offer sufficient background knowledge of the proposed scheme, these techniques are illustrated as follows. (The notations are listed in Table 1).

2.1 Paillier Homomorphic Cryptosystem

In 1999, Paillier proposed a probabilistic public-key cryptosystem [19] based on the composite residuosity class problem. Paillier's encryption scheme with fast decryption can be described as follows.

2.1.1 Key Generation Phase

Choose an RSA modulus $N = p \cdot q$, where p and q are large primes. Compute Carmichael's function taken on N, *i.e.*, $\lambda = \lambda(N) = lcm(p-1, q-1)$, and choose an element, $g \in \mathbb{Z}_{N^2}^*$, of an order divisible by αN for some α, where $1 \leq \alpha \leq \lambda$.

Now, the public key is $PK = (N, g)$, and the secret key is $SK = \alpha$.

2.1.2 Encryption Phase

The plaintext space is $\mathbb{Z}_N$. Given a plaintext $M < N$, choose $r \in \mathbb{Z}_N^*$ at random, and let the ciphertext be:

$$C = e_{PK}(M) = g^M r^N \bmod N^2. \quad (1)$$

2.1.3 Decryption Phase

The plaintext space is $\mathbb{Z}_{N^2}$. Given a ciphertext, $C < N^2$, get the plaintext:

$$M = d_{SK}(C) = \frac{L(C^\alpha \bmod N^2)}{L(g^\alpha \bmod N^2)} \bmod N, \quad (2)$$

where $L(\mu) = (\mu - 1)/N$.

Based on an appropriate complexity assumption, this system is semantically secure, and it is a trivially additive homomorphism over $\mathbb{Z}_N$, which leads to other identities as we require here:

$$d_{SK}(e_{PK}(M_1) \cdot e_{PK}(M_2) \bmod N^2) = (M_1 + M_2) \bmod N, \quad (3)$$

$$d_{SK}((e_{PK}(M_1))^k \bmod N^2) = (kM_1) \bmod N, \quad (4)$$

$$d_{SK}(e_{PK}(M_1) \cdot g^{M_2} \bmod N^2) = (M_1 + M_2) \bmod N, \quad (5)$$

where $M_1, M_2 \in \mathbb{Z}_N, k \in \mathbb{N}$.

2.2 Shamir's (t, w)-Threshold Secret Sharing

In 1979, Shamir developed a (t, w)-threshold secret sharing scheme [25] based on polynomial interpolation and the fact that a univariate polynomial $y = f(x)$ of degree $t-1$ is uniquely defined by t points, (x_i, y_i) with distinct x_i, for $i = 1, 2, \cdots, t$. The scheme can decompose one secret into w shadows, with t shadows required to recover the original secret, where $t \leq w$, but no group of $t-1$ shadows can do so. It consists of the following two phases:

2.2.1 Shadow Distribution Phase

The trusted dealer starts with a secret integer, $S \geq 0$, that is to be distributed among w users. Thus, the dealer:

1) Chooses a prime $P > max(w, S)$.

2) Randomly selects $t-1$ independent coefficients $a_1, a_2, \cdots, a_(t-1), 0 \leq a_i \leq P-1$, to constitute a random polynomial with $t-1$ degree over $\mathbb{Z}_P$,

$$f(x) = S + \sum_{j=1}^{t-1} a_j x^j \bmod P.$$

3) Chooses w distinct non-zero elements of $\mathbb{Z}_P$, denoted as $x_i, 1 \leq i \leq w$.

4) Computes $s_i = f(x_i) \bmod P, 1 \leq i \leq w$, and securely transfers the shadow, s_i, to user U_i, along with the public index x_i.

2.2.2 Secret Reconstruction Phase

Assume that users $U_{i_1}, U_{i_2}, \cdots, U_{i_t}$ pool their shadows to compute the secret S. Their shadows provide t distinct points $(x_{i_j}, s_{i_j})'s$, $1 \le j \le t$, which allow the computation of the coefficients of $f(x)$ by Lagrange interpolation. The secret, S, can be expressed as:

$$S = f(0) = \sum_{j=1}^{t} s_{i_j} c_{i_j} mod\, P,$$

where $c_{i_j} = \prod_{1 \le k \le t, k \ne j} \frac{x_{i_j}}{x_{i_k} - x_{i_j}} mod\, P, 1 \le j \le t.$

3 Related Works

By utilizing the redundancy of value representation in the Paillier cryptosystem, Wu *et al.* [27] proposed an encrypted, signal-based RDH for the scenario in which the extraction occurs after decryption. To embed a bit b, a pixel m is mapped to $2m + b$, *i.e.*, $e_{PK}(m)$ is changed to another encrypted value, $e_{PK}(2m + b)$.

There are three parties, *i.e.*, an image owner, a data-hider, and a receiver, corresponding to the three phases. The algorithm runs as follows. In the image encryption phase, for a pixel m, the image owner uses the Paillier cryptosystem to generate the ciphertext, $c = e_{PK}(m)$. In the data embedding phase, to embed a bit b_1, the data-hider sequentially computes:

$$\bar{c} = (c \cdot c) mod\, N^2$$

and

$$c' = \begin{cases} (\bar{c} \cdot e_{PK}(1)) mod\, N^2 & \text{if} \quad b_1 = 1 \\ \bar{c} & \text{if} \quad b_1 = 0 \end{cases},$$

which implies that $c' = e_{PK}(2m + b_1)$. When the Paillier modulus N is chosen to be sufficiently large to ensure that, in data extraction phase, $2m + b_1 < N$, *i.e.* $2m + b_1 mod\, N = 2m + b_1$, the receiver can obtain the correct values of the original pixel, m, and the hidden bit, b_1, by computing:

$$m = \lfloor d_{SK}(c')/2 \rfloor$$

and

$$b_1 = d_{SK}(c') - 2m.$$

The embedding of multiple bits can be accomplished iteratively. For example, if the second bit, b_2, is to be embedded into the encrypted value of the pixel m, based on the encrypted value, c', of the pixel with hidden bit, b_1, the data-hider sequentially computes:

$$\bar{c'} = (c' \cdot c') mod\, N^2$$

and

$$c'' = \begin{cases} (\bar{c'} \cdot e_{PK}(1)) mod\, N^2 & \text{if} \quad b_2 = 1 \\ \bar{c'} & \text{if} \quad b_2 = 0 \end{cases},$$

which implies that $c'' = e_{PK}(2(2m + b_1) + b_2)$.

Therefore, if one wants the embedding rate to reach μ bpp, μ iterations are required, and some room must be vacated for recording associated information, such as the number of iterations.

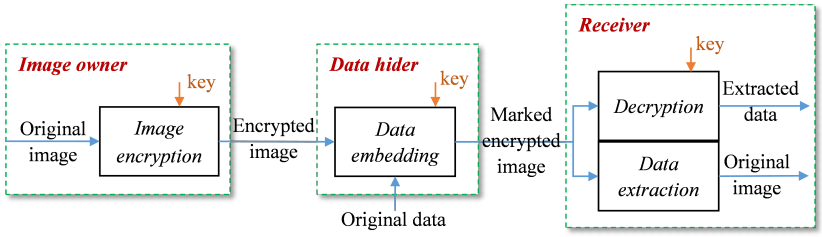
Similarly, Li *et al.* [14] encrypted the image pixel using the Paillier cryptosystem, and then they handled data embedding from the perspective of histogram shifting in the plain domain. First, for embedding one bit per pixel, the histogram of the host image is expanded by a factor of two, *i.e.*, from [0, 255] to [0, 511], so that the zero bins in the expanded histogram with odd numbers are vacated. Second, for embedding a bit, b, into the pixel, m, if the embedded bit b is 1, the corresponding unit of $2m$ in the expanded histogram shifts right by one step. The value $2m$ is processed in the plaintext image, the encryption of which can be obtained by computing $\bar{c} = (e_{PK}(m))^2 mod\, N^2$, and the encrypted value of the pixel m with the hidden bit is obtained by computing $c' = (\bar{c} \cdot g^b) mod\, N^2$. According to the additive homomorphism, c' is a valid encryption of $2m+b$, so reversibility is achieved. When one wants the embedding rate to be 1016 bits, the Paillier modulus N must be at least 1024 bits, and the expansion ratio of the pixel is 2^{1016}.Thus, the calculation in encryption domain is $\bar{c} = (e_{PK}(m))^{2^{1016}} mod\, N^2$.

4 A Reversible Data Hiding Scheme with Single-Receiver

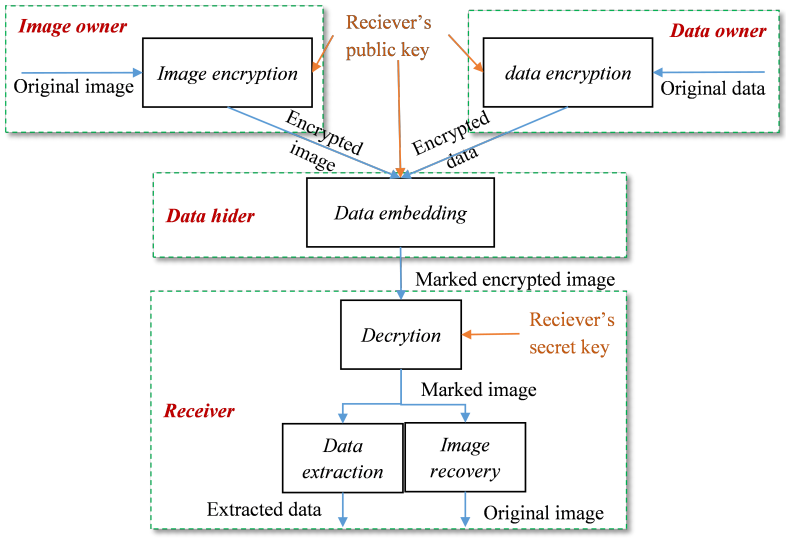
In this section, we present an RDH scheme using the Paillier cryptosystem for a single receiver. Figure 1(b) shows the flowchart of the process. There are five parties in the scheme, *i.e.*, a trusted dealer, an image owner, a data provider, a data processor, and a receiver. The trusted dealer generates the Paillier cryptosystem's public/private key pair, secretly sends a private key to the receiver, and broadcasts the public key to everyone. The image owner and the data provider use the Paillier cryptosystem with the receiver's public key to encrypt each pixel of the host image and the initial data, respectively, and the data provider, who does not know the host image, can modify the pixel values of the ciphertext to embed some additional data into the encrypted image. After receiving the encrypted image with the additional data embedded, the receiver, who has the private key of the cryptosystem, can execute decryption directly to get a marked image and then recover the hidden data and the host image perfectly.

4.1 Initialization Phase

When inputting the payload parameter, K, choose an RSA modulus, N,that is greater than K, such that $K' = \lfloor N/K \rfloor > 255$. Then the image space is $\{0, 1, \cdots, K'-1\}$, and the data space is $\{0, 1, \cdots, K-1\}$. Choose an element, $g \in \mathbb{Z}^*_{N^2}$, that has an order divisible by αN for some $1 \le \alpha \le \lambda(N)$.



(a) Sketch of the existing RDH scheme with public key cryptosystem [16]



(b) Sketch of the proposed RDH scheme with public key cryptosystem

Figure 1: Comparison between the existing scheme and the proposed RDH scheme

The public key $PK = (N, K, g)$ is broadcasted, and the secret key $SK = \alpha$ is sent secretly to the receiver.

4.2 Image Encryption Phase

Given the image $m < K'$, the image owner uses the receiver's public key PK to compute the ciphertext from Equation (1):

$$c_1 = e_{PK}(m) = (g^m r_1^N) mod\, N^2,$$

where $r_1 \in \mathbb{Z}_N^*$ is chosen randomly.

Then, the image owner sends the ciphertext, c_1, with the receiver's public key, PK, to the data processor.

4.3 Data Encryption phase

Given the data $b < K$, the data owner uses the receiver's public key, PK, to compute the ciphertext from Equation (1):

$$c_2 = e_{PK}(b) = (g^b r_2^N) mod\, N^2,$$

where $r_2 \in \mathbb{Z}_N^*$ is chosen randomly.

Then, the image owner sends the ciphertext, c_2, with the receiver's public key, PK, to the data processor.

4.4 Data Hiding Phase

To embed the hidden data, $c_2 < K$, into the ciphertext, c_1, which is encrypted by the reciever's public key, PK, the data processor computes:

$$c = (c_1 \cdot c_2^{k'}) mod\, N^2,$$

and sends the cipher, c, to the receiver.

4.5 Decryption and Extraction Phase

Given the cipher, c, the receiver first decrypts c with the secret key, SK, to get the marked message from Equation (2) as

$$m' = d_{SK}(c) = \frac{L(C^\alpha mod\, N^2)}{L(g^\alpha mod\, N^2)} mod\, N. \quad (6)$$

Then the host image can be obtained by:

$$m = m' mod\, K',$$

and the hidden data can be extracted by:

$$b = \lfloor m'/K' \rfloor.$$

Here we illustrate a simple numerical examples. Suppose the public key $PK = (N, K, g) = (15, 3, 16)$ and the secret key $SK = \alpha = 4$. Given the image pixel

$m = 3$, the image owner can encrypt it with a random number $r_1 = 2$ as: $c_1 = 16^3 \times 2^{15} \bmod 15^2 = 53$. Meanwhile, the data owner can encrypt the data $b = 1$ with a random number $r_2 = 4$ to obtain the encrypted data $c_2 = 16^1 \times 3^{15} \bmod 15^2 = 34$. While the data hider receive the encrypted image and the encrypted data, the hiding is done as $c = 53 \times 34^{\lfloor 15/3 \rfloor} \bmod 15^2 = 122$. When the receiver gets the cipher c, the marked message $m' = \frac{\left(122^4 \bmod 15^2 - 1\right)/15}{(16^4 \bmod 15^2 - 1)/15} \bmod 15 = 8$ is obtained first, then the host image pixel can be recovered as $m = 8 \bmod 5 = 3$, the hidden data can be extracted as $b = \lfloor 8/5 \rfloor = 1$.

5 A Multi-Receiver Reversible Data Hiding Scheme

In this section, a reversible data hiding scheme is proposed for sharing data among multiple receivers by combining the homomorphism property of Paillier encryption and polynomial interpolation. The aim of this scheme is to distribute both the image and the hidden data into multiple shadows prior to outsourcing them to the database center. This is necessary because the processing center will embed every data shadow into the responding image shadow to conduct a marked encrypted shadow for w receivers so that more than t receivers who collect their decrypted shadows can recover the host image and the plain data.

The proposed second scheme consists of five phases, *i.e.*, the initialization phase, the image partition and encryption phase, the data partition and encryption phase, the data embedding phase, and the decryption and reconstruction phase.

5.1 Initialization Phase

When the payload parameter, K, is input, the trusted dealer chooses an RSA modulus $N > K$, such that $K' = \lfloor N/K \rfloor > 255$. Then, the dealer selects a prime, $P > max(w, K, K')$ and chooses an element, $g \in \mathbb{Z}^*_{N^2}$, that has an order divisible by αN for some $1 \leq \alpha \leq \lambda(N)$. Choose w non-zero elements, $x_1, x_2, \cdots, x_w \in \mathbb{Z}_P$, randomly, and then x_i is distributed to the receiver R_i as her or his index.

The public key, $PK = (w, N, K, g)$, is broadcasted, while the secret key $SK = \alpha$ is sent secretly to the receivers.

5.2 Image Partition and Encryption Phase

1) Given the image $m < K'$, the image owner randomly selects $t-1$, independent coefficients, $a_{11}, a_{12}, \cdots, a_{1,t-1} \in \mathbb{Z}_P$, that define the random polynomial over $\mathbb{Z}_P$,

$$f_1(x) = \left(m + \sum_{j=1}^{t-1} a_{1j} x^j\right) mod\, P.$$

2) The image owner computes $s_{1i} = f_1(x_i) mod\, P, 1 \leq i \leq w$, and securely transfers the shadow, s_{1i}, to receiver R_i.

3) The image owner uses the receiver's public key, PK, to compute the cipher shadows by the Paillier cryptosystem,

$$c_{1i} = e_{PK}(s_{1i}),\ i = 1, 2, \cdots, w.$$

Then, the image owner sends the cipher shadow sequence, $(c_{11}, c_{12}, \cdots, c_{1w})$, to the data processor.

5.3 Data Partition and Encryption Phase

1) Given the hidden data $b < K$, the data provider randomly selects $t-1$ independent coefficients, $a_{21}, a_{22}, \cdots, a_{2,t-1} \in \mathbb{Z}_P$, that define the random polynomial over $\mathbb{Z}_P$,

$$f_2(x) = \left(b + \sum_{j=1}^{t-1} a_{2j} x^j\right) mod\, P. \tag{7}$$

2) The data provider computes $s_{2i} = f_2(x_i) mod\, P, 1 \leq i \leq w$, and securely transfers the shadow s_{2i} to receiver R_i.

3) The receiver uses her or his public key, PK, to compute the cipher shadows by the Paillier cryptosystem,

$$c_{2i} = e_{PK}(s_{2i}), i = 1, 2, \cdots, w. \tag{8}$$

Then, the receiver sends the cipher shadow sequence $(c_{21}, c_{22}, \cdots, c_{2w})$ to the data processor.

5.4 Data Embedding Phase

After obtaining the two cipher shadow sequences, $(c_{11}, c_{12}, \cdots, c_{1w})$ and $(c_{21}, c_{22}, \cdots, c_{2w})$, the data processor computes:

$$c_i = (c_{1i} \cdot (c_{2i})^{K'}) mod\, N^2, i = 1, 2, \cdots, w,$$

and then distributes the marked cipher shadow, c_i, to the receivers R_i, respectively, for $1 \leq i \leq w$.

5.5 Decryption and Reconstruction Phase

Assume that at least t receivers, $R_{i_1}, R_{i_2}, \cdots, R_{i_t}$, pool their shadows and use the receiver's private key, SK, to compute:

$$s'_{i_j} = d_{SK}(c_{i_j}), 1 \leq j \leq t, \tag{9}$$

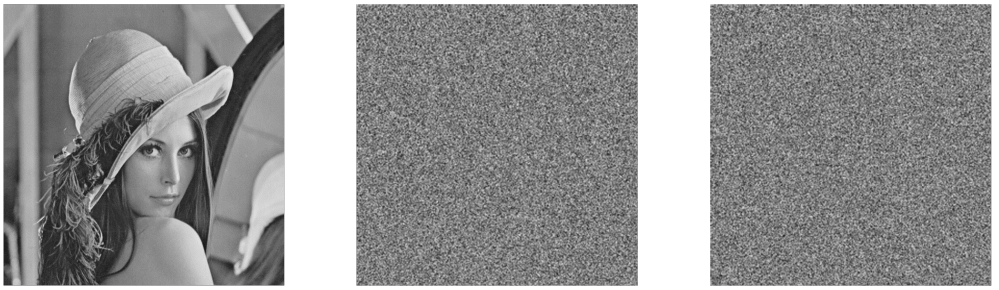
$$m' = \left(\sum_{j=1}^{t} s'_{i_j} \prod_{1 \leq k \leq t, j \neq k} \frac{x_{i_k}}{x_{i_k} - x_{i_j}}\right) mod\, P.$$

Then, the host image can be obtained by:

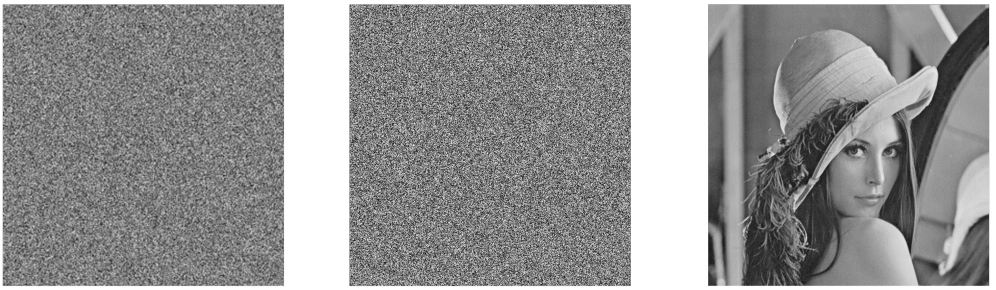
$$m = m' mod\, K', \tag{10}$$

and the hidden data can be extracted by:

$$b = \lfloor m'/K' \rfloor. \tag{11}$$



(a) Original image of "Lina" (b) Encrypted image (c) Encrypted image with 1 bpp



(d) Encrypted image with 100 bpp (e) Encrypted image with 1000 bpp (f) Recovered image

Figure 2: The original test image Lena, the encrypted image with different hidden data, and the recovered image

6 Performance Analysis

6.1 Verifying Reversibility

The reversibility of the single receiver RDH scheme in Section 4 can be verified easily in a theoretical analysis. Based on the homomorphic properties of the Paillier cryptosystem, which is shown in Equations (4)-(5), we know that Equation (6) is equivalent to $m' = m + K'b \bmod N$. Under the condition of $0 \le m \le K'-1$ and $0 \le b \le K-1$, we have $0 \le m + K'b \le K'K - 1 < N$, so that

$$(m + K'b) mod\, N = m + K'b,$$

and then, the original values of m_1 and m_2 can be recovered by Equations (10) and (11), respectively.

In the experimental analysis, we chose the modulus N that was 1024 bits in length, the original "Lena" gray-scale, 512×512 image, the encrypted image with no data embedded, the marked encrypted image at different embedding rates (1, 100, and 1000 bpp, respectively), and the perfectly recovered image are shown in Figure 2. The four images in Figures 2(b)-(e) were obtained by performing the arithmetic modulo 256 on the real encrypted images. The same result is also shown in Figure 3 for "Baboon". The test images are came from the USC-SIPI Image Database.

The reversibility of the multi-receiver RDH scheme presented in Section 5 was verified as follows. Assume that t receivers, $R_{i_1}, R_{i_2}, \cdots, R_{i_t}$, honestly pool their shadows. Similar to the analysis above, Equation (9) is equivalent to:

$$s'_{i_j} = (s_{1,i_j} + s_{2,i_j}K') mod\, N = s_{1,i_j} + s_{2,i_j}K', 1 \le j \le t,$$

and using Lagrange interpolation, we have:

$$\begin{aligned} m' &= \left(\sum_{j=1}^{t} s'_{i_j} \prod_{1\le k\le t, j\ne k} \frac{x_{i_k}}{x_{i_k}-x_{i_j}}\right) mod\, P \\ &= \left(\sum_{j=1}^{t} (s_{1,i_j} + s_{2,i_j}K') \prod_{1\le k\le t, j\ne k} \frac{x_{i_k}}{x_{i_k}-x_{i_j}}\right) \\ &= \left(\sum_{j=1}^{t} \left(s_{1,i_j} \prod_{1\le k\le t, j\ne k} \frac{x_{i_k}}{x_{i_k}-x_{i_j}}\right) + K' \right. \\ &\quad \left. \cdot \sum_{j=1}^{t} \left(s_{2,i_j} \prod_{1\le k\le t, j\ne k} \frac{x_{i_k}}{x_{i_k}-x_{i_j}}\right)\right) mod\, P \\ &= (f_1(0) + K' \cdot f_2(0)) mod\, P = m + K'b. \end{aligned}$$

Therefore, Equations (7) and (8) hold, *i.e.*, the host image can be recovered exactly and the hidden data can be extracted correctly.

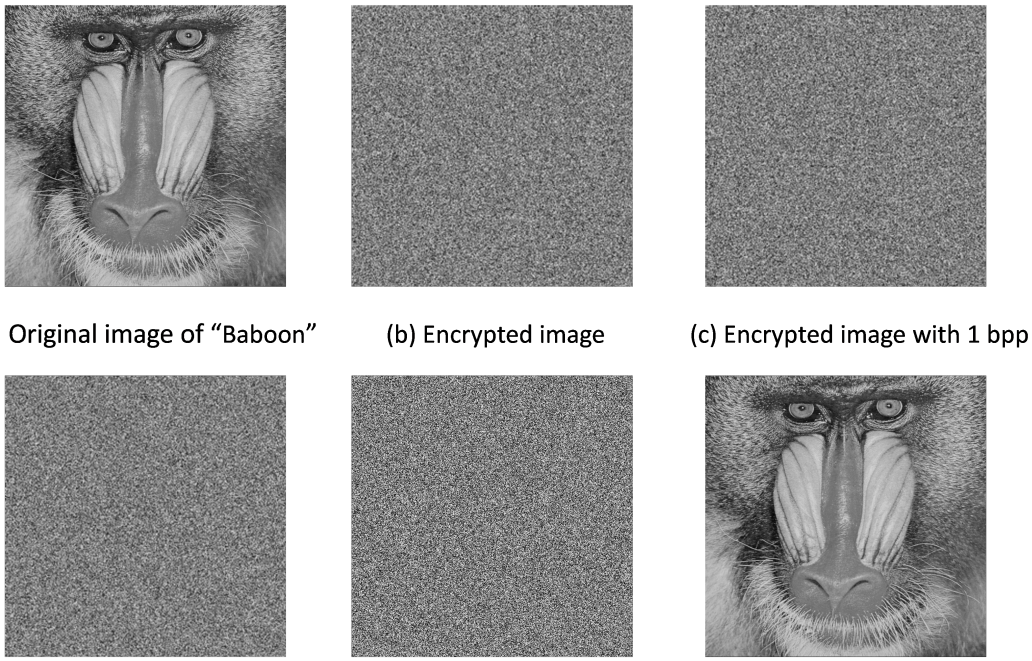
Figure 4 shows the illustration of the multi-receiver RDH scheme with (2,3)-secret sharing. Here we illustrate a small example for (3,5)-secret sharing reconstruction. Suppose that $P = 17$, $t = 3$, $w = 5$; and the i-th receiver's public index is $x_i = i$, for $1 \le i \le 5$. Suppose that three shares $(1, 8)$, $(3, 10)$ and $(5, 11)$ are pooled. Writing the polynomial $f(x)$ as $f(x) = a_0 + a_1 x + a_2 x^2$, then we have three linear equations in $\mathbb{Z}_{17}$:

$$\begin{cases} a_0 + a_1 + a_2 = 8 \\ a_0 + 3a_1 + 9a_2 = 10 \\ a_0 + 5a_1 + 8a_2 = 11 \end{cases}$$

This system has a unique solution in $\mathbb{Z}_{17}$: $a_0 = 13$, $a_1 = 10$ and $a_2 = 2$. Therefore the secret key is $f(0) = a_0 = 13$.

6.2 Embedding Capacity

The embedding capacity depends on the payload parameter, K, and up to $\lfloor log_2 K \rfloor$ bits can be hidden per pixel



(a) Original image of "Baboon" (b) Encrypted image (c) Encrypted image with 1 bpp

(d) Encrypted image with 100 bpp (e) Encrypted image with 1000 bpp (f) Recovered image

Figure 3: The original test image Baboon, the encrypted image with different hidden data, and the recovered image

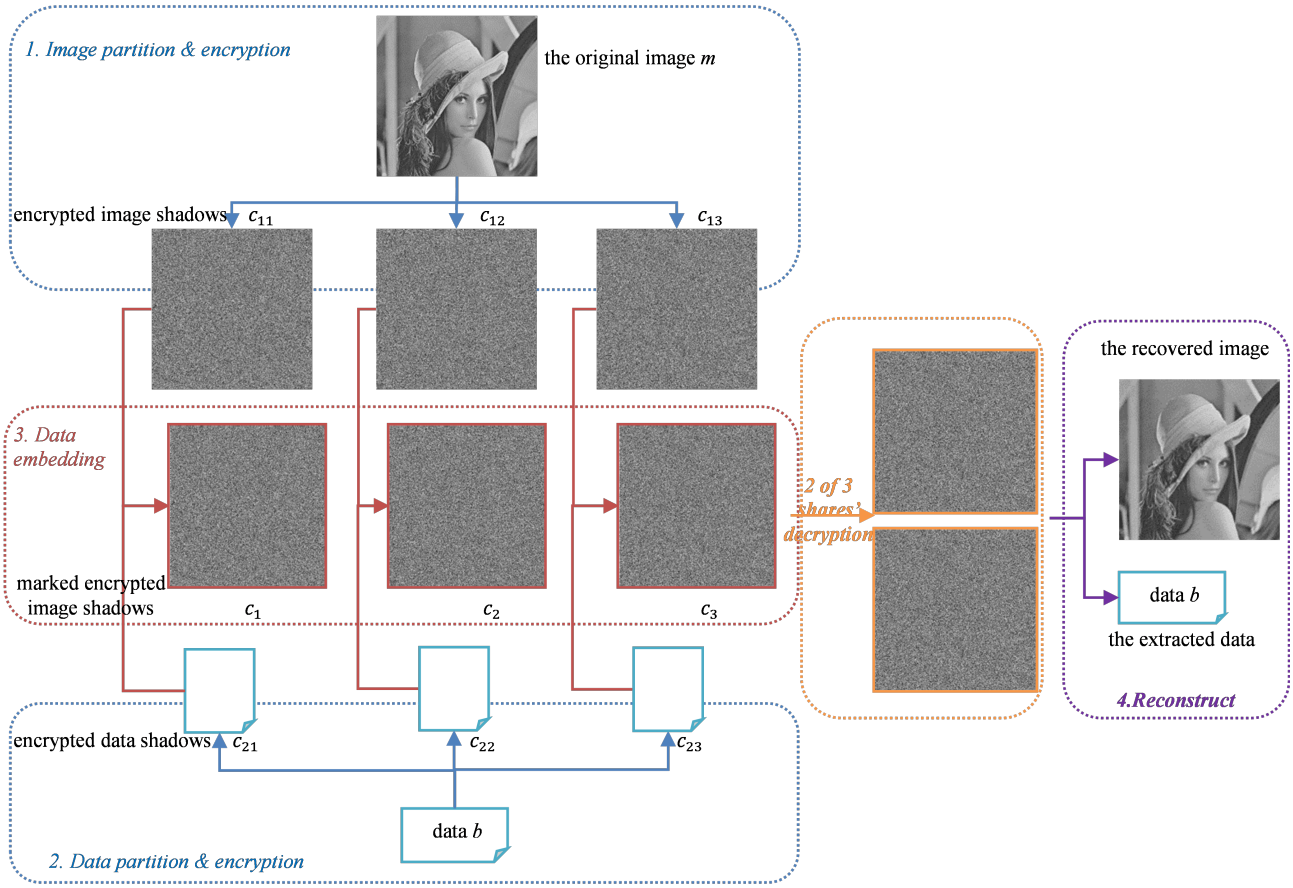


Figure 4: The illustration of the multi-receiver RDH scheme with (2, 3) - secret sharing

Table 2: Performance comparison

Method	Cryptosystem	Embedding capacity(bpp)	preprocess before encryption	Separability of data encryption & embedding
Ma *et al.* [16]	Stream cipher	0.5	yes	no
Zhang *et al.* [31]	Paillier encryption	< 1	no	no
Wu *et al.* [27] ($\|N\| = 1024$)	Paillier encryption	1016	no	no
Li *et al.* [14] ($\|N\| = 1024$)	Paillier encryption	1016	yes	no
Singh *et al.* [26]	secret sharing	≤ 2	no	no
Proposed Scheme ($\|N\| = 1024$)	Paillier encryption	$\|K\| \in [1, 1016]$	no	yes

and perfectly extracted with the appropriate modulus. As a grey-level pixel value from 0 to 255 can be represented with 8 bits, when a big modulus, N, with the bit length of 1024 was used, one pixel can embed up to 1016 bits, even when the length of N is only 9 bits, and 1 bit per pixel can be embedded and correctly extracted. The size of the modulus is related to the scale of the value expansion and the security of the scheme, so parameters can be chosen adaptively by the trade off between efficiency and security. The performance of the proposed algorithm was compared with those of several other algorithms [14, 16, 26, 27, 31], as shown in Table 2. When an 8-bit pixel value was encrypted into a 2048-bit big integer for N with 1024 bits in the Paillier cryptosystem, the embedding capacities of the proposed algorithms is much higher than those in [16,26,31]; although [27] and [14] attained the same capacity, the former must conduct data hiding 1016 iteratively, and the latter must perform extra processing of the images before encryption.

7 Conclusions

In this paper, we proposed two RDH schemes with large embedding capacity, *i.e.*,

1) One that is suitable for a single receiver;

2) One for multiple receivers.

These two schemes have the following common features:

1) Compared to the traditional scheme, we do the role separation between the data provider and data-hider for more application scenarios, and the hidden data also are transmitted in encrypted form;

2) The high embedding rate can be achieved adaptively according to requirements, ranging from 1 bpp to even more than 1016 bpp, which is irrelevant to the pixel distribution of the test image;

3) The schemes do not require an extra processing step before encryption;

4) The embedding rate is independent of the pixel distribution of different natural images;

5) Both the encryption key and the data hiding key are the receiver's public key, and the extraction of the data is done after decrypting the marked encrypted image with the corresponding private key.

In addition, the multi-receiver RDH scheme distributes trust among several receivers, the marked, encrypted image is shared among w receivers, and the host image and hidden data cannot be extracted unless t or more receivers cooperate. It was assumed that all of the receivers have the same private key, and this inflexibility may be improved in the future work by considering multi-secret sharing or proxy encryption.

Acknowledgments

This work was partly supported by the Natural Science Foundation of Fujian Province, China (Grant No. 2018J01537), the Education and Scientific Research Project for Young Middle-aged Teachers of Fujian Province, China (Grant No. JAT190314), and the Science and Technology project of Xiamen Municipal (Grant No. 3502Z20173028). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] Abd-Eldayem and M. Mohamed, "A proposed security technique based on watermarking and encryption for digital imaging and communications in medicine," *Egyptian Informatics Journal*, vol. 14, pp. 1–13, Mar. 2013.

[2] A. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Transactions on Information Theory*, vol. 30, pp. 208–210, Mar. 1983.

[3] J. C. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret," in *Advances in Cryptology (CRYPTO'86)*, pp. 251–260, Aug. 1986.

[4] K. Bharanitharan and C. C. Chang, H. R. Yang, and Z. H. Wang, "Efficient pixel prediction algorithm for reversible data hiding," *International Journal of Network Security*, vol. 18, pp. 750–757, July 2016.

[5] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of American Federation of Information Processing Societies National Computer Conference (AFIPS'79)*, pp. 313–317, June 1979.

[6] X. Cao, X. Wei L. Du, D. Meng, and X. Guo, "High capacity reversible data hiding in encrypted images by patch-level sparse representation," *IEEE Transactions on Cybernetics*, vol. 46, pp. 1132–1143, Mar. 2016.

[7] C. C. Chang, Y. H. Huang, and T. C. Lu, "A difference expansion based reversible information hiding scheme with high stego image visual quality," *Multimedia Tools and Applications*, vol. 76, pp. 12659–12681, May 2017.

[8] C. C. Chang, C. C. Lin, C. H. Lin, and Y. H. Chen, "A novel secret image sharing scheme in color images using small shadow images," *Information Sciences*, vol. 178, pp. 2433–2447, June 2008.

[9] Y. Chen, C. Shiu, and G. Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1164–1170, 2014.

[10] M. Chen X. Zeng G. Biao, Z. Chen and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Transactions on Information Forensics and Security*, vol. 5, pp. 187–193, Mar. 2010.

[11] S. F. Chiou, Y. C. Lu, I-En Liao, and M. S. Hwang, "An efficient reversible data hiding scheme based on SMVQ," *Imaging Science Journal*, vol. 61, no. 6, pp. 467–474, 2013.

[12] W. Hong, T. Chen, J. Chen, Y. Kao, H. Wu, and M. Wu, "Reversible data embedment for encrypted cartoon images using unbalanced bit flipping," in *Proceedings of the 4th International Conference in Swarm Intelligence*, pp. 208–214, June 2013.

[13] M. Khosravi and M. Yazdi, "A lossless data hiding scheme for medical images using a hybrid solution based on ibrw error histogram computation and quartered interpolation with greedy weights," *Neural Computing and Applications*, vol. 30, pp. 2017–2028, Oct. 2018.

[14] M. Li and Y. Li, "Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding," *Signal Processing*, vol. 130, pp. 190–196, Nov. 2017.

[15] L. Liu, C. C. Chang, and A. Wang, "Reversible data hiding scheme based on histogram shifting of n-bit planes," *Multimedia Tools and Applications*, vol. 75, pp. 11311–11326, Sep. 2016.

[16] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics Security*, vol. 8, pp. 553–562, Mar. 2013.

[17] R. J. McEliece and D.V. Sarwate, "On sharing secrets and reed-solomon codes," *Communications of the ACM*, vol. 24, pp. 583–584, Sep. 1981.

[18] M. Mignotte, "How to share a secret," in *Proceedings of the Workshop on Cryptography*, pp. 371–375, Mar. 1982.

[19] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology*, pp. 223–238, May 1999.

[20] W. Puech, M. Chaumont, and O. Strauss, "A reversible data hiding method for encrypted images," in *Proceedings of SPIE*, vol. 6819, pp. 1–9, Mar. 2008.

[21] C. Qin and Y. C. Hu, "Reversible data hiding in VQ index table with lossless coding and adaptive switching mechanism," *Signal Processing*, vol. 129, pp. 48–55, Dec. 2016.

[22] C. Qin and X. Zhang, "Effective reversible data hiding in encrypted image with privacy protection for image content," *Journal of Visual Communication and Image Representation*, vol. 31, pp. 154–164, Aug. 2015.

[23] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–179, 1978.

[24] H. Rostami, M. Khosravi and S. Samadi, *Enhancing the Binary Watermark-Based Data Hiding Scheme Using an Interpolation-Based Approach for Optical Remote Sensing Images*, 2019. DOI: 10.4018/978-1-5225-7033-2.ch014.

[25] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, pp. 612–613, Nov. 1979.

[26] P. Singh and B. Raman, "Reversible data hiding based on shamir's secret sharing for color images over cloud," *Information Sciences*, vol. 422, pp. 77–97, Jan. 2018.

[27] H. T. Wu, Y. M. Cheung, and J. W. Huang, "Reversible data hiding in paillier cryptosystem," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 765–771, Oct. 2016.

[28] X. Wu and W. Sun, "High-capacity reversible data hiding in encrypted images by prediction error," *Signal Processing*, vol. 104, pp. 387–400, Nov. 2014.

[29] X. Wu, J. Weng, and W. Yan, "Adopting secret sharing for reversible data hiding in encrypted images," *Information Sciences*, vol. 143, pp. 269–281, Feb. 2018.

[30] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, pp. 255–258, Apr. 2011.

[31] X. Zhang, J. Wang, Z. Wang, and H. Cheng, "Lossless and reversible data hiding in encrypted images with public key cryptography," *IEEE Transactions on Circuits Systems Video Technology*, vol. 26, pp. 1622–1631, Sep. 2016.

[32] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," *IEEE Transactions on Circuits Systems Video Technology*, vol. 26, pp. 441–452, Mar. 2016.

Biography

Hefeng Chen received the B.S. and M.S. degrees in mathematics from Xiamen University, China, in 2005 and 2008, respectively, and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2016. She was a visiting scholar with Feng Chia University, Taiwan, in 2018. She is currently an assistant professor with the Computer Engineering College, Jimei University, Xiamen, China. Her current research interests include cryptography and information hiding.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. He served in National Chung Cheng University from 1989 to 2005. Since February 2005, he has been a Chair Professor with Feng Chia University. He is a Fellow of IEEE and a Fellow of IEE, UK. His current research interests include computer cryptography, image compression, and data structures.

Kaimeng Chen received the B.Sc. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2010 and 2016, respectively. He was a visiting scholar with Feng Chia University, Taiwan, in 2018. He is currently an assistant professor with the Computer Engineering College, Jimei University, Xiamen, China. He is a principle investigator of a project of the National Science Foundation of Fujian Province, China. His research interests include data hiding, image processing, databases on new hardware, hybrid storage, and non-volatile memory technology.

Application of Novel Gabor-DCNN into RGB-D Face Recognition

Yuanyuan Xiao[1,2] and Xiaoyao Xie[2]
(Corresponding author: Xiaoyao Xie)
School of Computer Science and Technology, Guizhou University[1]
Huaxi District, Guiyang, China
Key Laboratory Of Information and Computing Science of Guizhou Province, Guizhou Normal University[2]
Yunyan District, Guiyang, China
(Email: xyx@gznu.edu.cn)
(Received Mar. 17, 2019; Revised and Accepted Aug. 4, 2019; First Online Dec. 13, 2019)

Abstract

In this paper, we propose a novel approach, named Gabor-DCNN, applied for face recognition technology of two modalities RGB-D images, which can extract the features through Gabor transform of images and deep convolutional neural network. Gabor transform can capture the salient visual properties with spatial frequencies and local structural features of different directions in a local area of images and enhance the object representation capability. Deep convolutional neural networks could automatically learn essential features from images compared with traditional methods. The most significant features can be obtained through the Gabor-DCNN. The final features expression of the face is performed by feature fusion of two modalities images. The experimental results indicate that the algorithm achieves much better performance than some state-of-the-art methods in terms of recognition rates on the EURECOM data set. This research provides an effective method for multiple modal face recognition under complex conditions.

Keywords: Deep Convolutional Neural Network; Face Recognition; Gabor Transform; RGB-D

1 Introduction

With the rapid development of the Internet, the technology of identity authentication based on face recognition has been extensively applied to different kinds of fields such as mobile payment, entrance guard system, and so on. Traditional algorithms of face recognition mostly use two-dimensional images, which can not effectively prevent information forgery and cope with changes such as illumination, posture, and expression. 3D images can capture more information, thus enabling higher preservation of facial detail under varying conditions [8,21]. Multidimensional spatial data acquisition equipment has been used in computer vision for many years, while the high cost limits their usage in large scale applications. With the development of sensor technology, RGB-D devices, such as Microsoft's Kinect sensor, have been widely initially used for gaming and later became a popular device for computer vision and high-quality data can be acquired easily [31]. RGB-D data obtained from binocular cameras take advantage of color images that provide appearance information of an object and Depth images [10, 17, 28] that are immune to the variations in color, illumination, rotation angle, and scale. The Depth image is a characterization of geometry, which contains accurate information related to the distance. Each pixel value of Depth image is the precise range between the sensor and the object. In recent years, a tremendous increasing number of RGB-D data is applied for different fields including object recognition, scene classification, hand gesture recognition, pose estimation, and 3D-simultaneous localization and mapping [13,23,26,27]. Depth images are also used in privacy protections [9].

In face recognition fields, there are many methods to extract features for two-dimensional images such as principal component analysis (PCA), linear discriminant analysis (LDA), which are statistic methods and can efficiently recognize targets. Many other effective approaches have been widely used such as local binary pattern (LBP), the histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT) and so on. The visual features from single RGB images are not entirely handing against changes such as illuminance change. RGB-D images may potentially enhance the robustness of the feature descriptor to overcome these problems [22, 31]. There are some algorithms about the face recognition of RGB-D. Entropy and saliency are used to obtain feature maps [8], and then features are extracted by the histogram of oriented gradient (HOG), which strengthens the role of RGB images and weakens that of Depth maps. The method of 3D Local Binary Pattern (3DLBP) is used to extract fea-

tures [18], which is mainly based on histogram statistics of features spectrum. Since 2012, convolutional neural networks (CNNs) have recovered rapidly [14, 15], conquered most fields of computer vision, and are booming. CNN can be utilized as a special feature extractor to acquire stable feature representation or one of deep learning methods that combines feature extractor and classifier. The Gabor-DCNN, for the task of RGB-D face recognition, combines Gabor transform [19, 20] that is used for strengthening the texture information with the deep convolutional neural network (DCNN) [15], which is introduced for face recognition due to their excellent performance in computer vision. Through constructing the Gabor filters and a series of operations such as convolution and pooling, the correlation of images' data is fully excavated. Gabor feature maps that utilize Gabor transform to get directional properties and spatial density as the input data of the deep convolutional neural network are used to extract more salient features by fully training the network.

The proposed approach is evaluated in experiments with comparisons to some state-of-the-art methods on EURECOM data set [21]. In the domain of computer vision, while the convolutional neural network has been widely applied for face recognition and object categorization, the proposed approach could further enhance the capacity of feature representation by Gabor transform of images. As the experimental results demonstrate, the method is so robust as to facial expression variations.

The rest of the paper is organized as follows. The algorithm of Gabor-DCNN applied for RGB-D face recognition is described in Section 2. The experimental evaluations of the proposed approach with the comparisons to some state-of-the-art methods are reported in Section 3. The conclusions are drawn and future perspectives are given in Section 4.

2 The Algorithm of Gabor-DCNN

The Gabor-DCNN is presented based on RGB-D images of multi-sensor, which combines the Gabor transform [25] of images with the constructed deep convolutional neural network. Gabor transform, which can extract relevant features of images in different scales and orientations, is a robust face classification descriptor, and Gabor feature maps are widely applied as robust feature representations of images. The deep convolutional neural network as one of deep learning methods can get higher accuracy in image classification by a series of operations. The proposed algorithm can extract the most remarkable features and improve the expressive ability of the object, which is comprised of five major steps in the following subsections:

2.1 Data Preprocessing

RGB-D images obtained from binocular cameras include two types: RGB images and Depth maps, and there is a one-to-one match between them. The pixel values of the Depth image, which reflects the geometric shape of the visible surface of the object directly, represents the range between the object surface and the image capture device in the scene. The holes, existing in Depth images, which are caused by light and distance, are similar to the background, and then linear interpolation is utilized to fill the holes in three channels respectively. Viola-Jones detection [2] is carried out to acquire face regions and preserves detection frames, utilized to crop face region of the Depth map. Face detection is one of the essential techniques used in automatic processing by the computer, playing an essential role in face recognition, the aim of which is to acquire the human face information from the background contained in the digital images.

2.2 Gabor Feature Maps

Gabor feature maps can be obtained through Gabor transform, which is used to solve the problem that signals cannot be analyzed locally by Fourier transform and Gabor filters. The physical meaning of Fourier transform is to transform the gray distribution function of the image into the frequency distribution function of the image, which reflects the statistical characteristics of frequency signals, and Gabor transform is a windowed Fourier transform, the windows function of which is Gaussian function. Generally, the Gabor feature maps can be acquired in two steps:

- Constructing Gabor filters;
- Gabor transform of images by Gabor filters.

Gabor filters, which are similar to the 2D receptive field profiles of simple cells of the mammalians' visual cortex, have excellent spatial locality and directional selectivity, can grasp the local features of spatial frequency and structure in multiple directions of the image [4]. Gabor feature maps can be gotten by a set of Gabor filters, which capture the salient visual properties in images [21]. The Gabor filter is defined as follows:

$$g(x,y;\omega,\sigma,\theta) = \frac{1}{2\pi\sigma^2}\cdot exp\{-\frac{x^2+y^2}{2\sigma^2}\} \cdot exp\{j\omega(xcos\theta + ysin\theta)\}. \quad (1)$$

Here $j=\sqrt{-1}$, θ controls the orientation of the Gabor filters, $\theta = \frac{\pi\cdot m}{M}$ with $m = 0, \cdots, M-1$, and M stands for the number of orientations; ω represents frequency, $\omega = \frac{\omega_{max}}{f^n}$ with $n = 0, \cdots, N-1$, and N denotes the number of frequencies; f is the interval factor of frequency; σ is the standard deviation of the Gaussian envelope, which determines the width of Gaussian windows. The frequency ω and envelope σ control the sparsity of the Gabor feature maps.

As can be seen from Formula (1), Gabor filters are subjected to three variable: θ, ω, and σ. A specific value of Gaussian envelope: σ, will be set to 2π. Hence, the

Gabor filters can be constructed only by setting parameters: frequency and direction. We choose the maximum frequency: $\omega_{max} = \frac{\pi}{2}$, due to large-scale images usually choose smaller frequencies from these papers [3, 5, 16, 24]. Also, the Gabor feature maps with five frequencies and eight orientations are the best. The interval factor of frequency is set to $\sqrt{2}$. According to the requirement, the parameters of the Gabor filter are shown in Table 1.

Table 1: The parameters list of Gabor filters

Parameters	Values
$\omega_i(i = 1, \cdots, 5)$	$\frac{\pi}{2}, \frac{\pi}{2\sqrt{2}}, \frac{\pi}{4}, \frac{\pi}{4\sqrt{2}}, \frac{\pi}{8}$
$\theta_j(j = 1, \cdots, 8)$	$0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \frac{\pi}{2}, \frac{5\pi}{8}, \frac{3\pi}{4}, \frac{7\pi}{8}$

The Gabor transform is defined as its convolutions of an image and the Gabor filters; the formal is shown as follows:

$$G(x, y) = I(x, y) * g(x, y; \omega, \sigma, \theta). \quad (2)$$

Where $I(x, y)$ stands for the images, the symbol " $*$ " denotes the convention operator, and $G(x, y)$ represents the robust feature maps obtained by Gabor transform. It is worth noting that images should be converted into gray images before Gabor transform. The features maps acquired by eight directions of Gabor filters will be combined into an 8-channel image at the same frequency, prepared for the input images of the DCNN constructed in the next subsection.

Consequently, through a series of transformation and channels fusion of images according to the rules we just mentioned, the RGB/Depth images with three channels will be converted into 8-channel images with five different frequencies respectively. The Gabor feature maps at five frequencies with eight channels can be represented as follows:

$$H_{rgb}(i) = CF_{\omega_i}(G_{\theta_j}(RGB)), \quad (3)$$
$$H_{depth}(i) = CF_{\omega_i}(G_{\theta_j}(Depth)). \quad (4)$$

Where $i \in [1, 5], j \in [1, 8]$. $CF(\cdot)$ represent the fusion of 8-channel images, which is computed by setting parameters according to Table 1. The $H_{rgb}(i)$ indicates that they are Gabor feature maps with the i-th frequency fused images of RGB images, and the $H_{depth}(i)$ indicates that they are Gabor feature maps with the i-th frequency fused images of Depth maps. Fused images aim to reduce the computational cost of feature extraction, and the most remarkable features by training the deep convolutional neural network.

2.3 Feature Extraction

In this paper, we adopt a deep convolutional neural network model to identify 8-channel fused images, which improves LeNet-5 [7] model. The LeNet-5 is a convolutional neural network including two convolution layers, two down-sampling layers, and three fully-connected layers, which is used to identify handwritten digits. The improved model, designed in Table 2, is used to recognize the fused images that are processed by Gabor transform. The improved model and Gabor transform of images can get more remarkable features.

As shown in Table 2, Conv5-6 stands for convolution layer with six filters of 5×5 size; S=2 represents stride and the value of S is 2; Max-pooling denotes that max-value is selected in pooling layer. FC represents full connection layer. The DCNN model is a six-layer deep convolutional neural network, which consists of five CS (convolution and down-sampling, CS) layers and a full connection layer. The abstract features will be extracted by convolution layers of the networks, and will be more powerful representation with the increase of convolution layers [6, 30]. The full connection layer of our network is designed to accomplish the number of object classification task and the numclass stands for the number of classes.

From Table 2, we can find that the differences between LeNet-5 and the DCNN model, which are summarized as follows. The constructed DCNN has five CS layers and a full connection layer. A CS layer stands for a convolution layer, a BN, a ReLU, and a Max-pooling layer. A ReLU, selected in the improved model, is adopted as the activation function. The activation function of the LeNet-5 model is sigmoid. Batch normalization (BN) and Dropout are used to optimize neural networks [11, 12]. The usage of BN can not only efficiently improve the training speed of the network but also enhances the generalization ability of the model. Dropout is a simple but effective regularization technique, which makes some nodes of the network not only work stochastically but also improves the generalization ability of the neural network in the training process of the neural network.

The CNN model is used to extract features. The features are extracted by the DCNN model we constructed as follows:

$$F_{rgb}(i) = C(H_{rgb}(i)), \quad (5)$$
$$F_{depth}(i) = C(H_{depth}(i)). \quad (6)$$

Where $i \in [1, 5]$, and the DCNN model is represented as $C(\cdot)$, utilized as a feature extractor. The most prominent features are selected by using a series of convolution layers and Max-pooling layers under five different frequencies. Five kinds of features obtained by extracting different 8-channel images are concatenated into a single feature vector, which represents features of a single modality. Both RGB images and Depth maps are adopted in the same way to extract features. Consequently, features of RGB-D are linked to an extended vector F as final feature expression, which is provided as the input data to a multi-class classifier. The features F are respresented as follows:

$$F = [F_{rgb}, F_{depth}]. \quad (7)$$

Table 2: Operations, dimensions and parameters list of DCNN model

Layers	Parameters
Input Layer	$256 \times 256 \times 8$
CS1 Layer	Conv5-6 (S=1), BN, ReLU, Max-pooling (S=2)
CS2 Layer	Conv5-12 (S=1), BN, ReLU, Max-pooling (S=2)
CS3 Layer	Conv6-12 (S=1), BN, ReLU, Max-pooling (S=2)
CS4 Layer	Conv5-24 (S=1), BN, ReLU, Max-pooling (S=2)
CS5 Layer	Conv5-48 (S=1), BN, ReLU, Max-pooling (S=2), Dropout
FC Layer	numclass

2.4 Pattern Classification

The selection of classifiers largely determines the efficiency of pattern recognition. Some classifiers such as Nearest Neighbour (NN), Random Decision Forests (RDFs), and Support Vector Machine (SVM) have been applied extensively into pattern classification. SVM, which is a robust supervised algorithm method of machine learning and extremely computationally accurate during probe identification, is chosen as a classifier in the proposed approach. SVM can also be applied for regression and classification, the non-linear relationship of which can get between data and features when the sample size is small and medium. Therefore, SVM is very suitable for the classification of features in this study. The category labels can be obtained by SVM.

2.5 The Computation of Accuracy

The category labels are compared with the input labels and the recognition rate (or accuracy) is calculated as follows:

$$Accuracy = \frac{Sum(L(x) == L_{input}(x))}{Numel(L_{input}(x))} \times 100\%. \quad (8)$$

Here, $L(x)$ stands for the category labels obtained by classifier; $L_{input}(x)$ represents the input label; the symbol " == " denotes object equality, and returns logical 1 (true) only where $L(x_i)$ and $L_{input}(x_i)$ are equal, the result of " $L(x) == L_{input}(x)$ " is a vector that contains 1 or 0. $Sum(\cdot)$ is utilized to calculate the sum of the vector elements, the values of which are 1. $Numel(\cdot)$ returns the number of elements, which exists in $L_{input}(x)$. To summarize, the method of Gabor-DCNN is sketched in Algorithm 1 as follows:

Algorithm 1 Gabor-DCNN

Input: The RGB-D images, I_{rgb} stands for the RGB image, I_{depth} denotes the Depth map, and the meaning of symbol " ∀ " is on behalf of all the thing.

Output: *Category Labels*, *Accuracy*

1: Begin

2: The I_{rgb} and I_{depth} are converted to grayscale images respectively at first as:

$$\begin{aligned} Ig_{-rgb} &= grayscale(I_{rgb}), \\ Ig_{-depth} &= grayscale(I_{depth}). \end{aligned}$$

3: Gabor filters are constructed according to the specific frequency, Gaussian envelope, and orientation. Details about the parameters are shown in Table 1, and a set of Gabor filters are calculated as follows by Equation (1):

$g_{ij} = \frac{1}{2\pi\sigma^2} \cdot exp\{-\frac{x^2+y^2}{2\sigma^2}\} \cdot exp\{j\omega_i(xcos\theta_j + ysin\theta_j)\}$,

$g = \forall g_{ij}$, where $i \in [1,5]$, and $j \in [1,8]$.

4: Gabor featur maps are calculated by Equation (2) for Ig_{-rgb} and Ig_{-depth} respectively as:

$$\begin{aligned} Grgb_{ij} &= Ig_{-rgb} \cdot g_{ij}, \\ Grgb &= \forall Grgb_{ij}, \\ Gdepth_{ij} &= Ig_{-depth} \cdot g_{ij}, \\ Gdepth &= \forall Gdepth_{ij}. \end{aligned}$$

Where $i \in [1,5]$ and $j \in [1,8]$.

5: The images with same frequency are combined into an 8-channel image based on Equation (3) and Equation (4) respectively, and then $Hrgb$ and $Hdepth$ stand for Gabor feature maps of RGB images and Depth maps by a series of image processing respectively:

$$\begin{aligned} H_{rgb}(i) &= CF_{\omega_i}(Grgb_{ij}), \\ Hrgb &= \forall H_{rgb}(i), \\ H_{depth}(i) &= CF_{\omega_i}(Gdepth_{ij}), \\ Hdepth &= \forall H_{depth}(i). \end{aligned}$$

Where $i \in [1,5]$ and $j \in [1,8]$.

6: The constructed DCNN model, which is a feature extractor and is represented as $C(\cdot)$, is used to extract features from fused images at a certain frequency of single modality based on Equation (5) and Equation (6), the features of two modalities images are expressed respectively as follows:

$$\begin{aligned} F_{rgb}(i) &= C(H_{rgb}(i)), \\ F_{depth}(i) &= C(H_{depth}(i)), \text{ where } i \in [1,5]. \\ F_{rgb} &= \forall F_{rgb}(i), \\ F_{depth} &= \forall F_{depth}(i), \text{ where } i \in [1,5]. \end{aligned}$$

7: The features vector F denotes the final features expression of two modalities by concatenating to reduce

the number of vectors from two to a single vector by Equation (7) as follow:

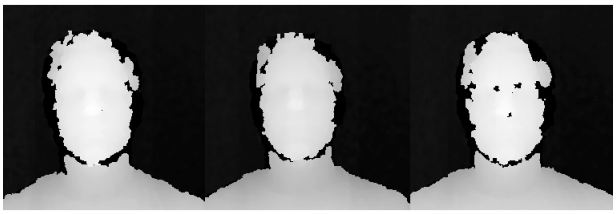
$$F = [F_{rgb}, F_{depth}].$$

8: SVM is chosen to obtain category labels and the accuracy is calculated according to Equation (8).
9: End

3 Experimental Results and Analysis

In order to validate the proposed method, we apply the proposed algorithm to RGB-D face images of EURECOM data set. Face recognition technology is a wonderfully challenging theme in the field of computer vision and pattern recognition [3, 29]. 936 images, pertaining to 52 people in EURECOM data set, are captured in the two-time series: Session 1 (S1) and Session 2 (S2), and some samples can be seen in Figure 1.

(a) RGB images



(b) Depth maps

Figure 1: The samples of the EURECOM data set

EURECOM data set has higher simulation degree with covariant variables such as pose, illumination, and occlusion. Since the EURECOM data set does not divide the testing set and the training set. It is necessary to balance the samples and divide the training set and the testing set before the experiments.

3.1 Features Extraction of Two Modalities Images

In order to test the effectiveness and the robustness of Gabor-DCNN, we will achieve the experiments for features extraction of two modalities images. Forty Gabor filters will be constructed according to the parameters in Table 1 at first, and then forty Gabor features maps of each modality can be processed by Gabor transform. The images with the same frequency will be fused into eight-channel images. Hence, ten kinds of Gabor feature maps under five different frequencies, which are used as the input of the constructed DCNN model. The recognition rates are shown in Figure 2 by fully training the DCNN model.

From Figure 2, we can find: the accuracies of the RGB images perform well, which can run at more than 96%; the recognition rates of the Depth maps are slightly lower than that of the RGB images, which are about 90%. To further investigate the performance of the proposed method, we also have tested RGB-D on other models such as LeNet-5, MT-LeNet-5, and DeepID2. LeNet-5 model is utilized to identify single-channel images such as grayscale images, we should convert two kinds of images into grayscale images as the input of the LeNet-5 model before training the model; MT-LeNet-5 means modified three-channel LeNet-5, used to achieve classification task of 3-channel images, which is different from LeNet-5 at first convolution layer; DeepID2 is also a 3-channel convolutional neural network, which is used to identify 3-channel images. The recognition rates of different network models are shown in Table 3.

Table 3: The comparison of recognition rates in different network models

Input images	Accuracy(%)		
	LeNet-5	MT-LeNet-5	DeepID2
RGB	88.78	89.42	88.46
Depth	57.37	60.58	40.38

As can be seen from the Table 3, the first column represents the input of data; the second column stands for recognition rates of LeNet-5; The third column denotes the accuracy of MT-LeNet-5; The fourth column stands for the recognition rates of DeepID2. Both RGB images and Depth maps are classified by MT-LeNet-5 and DeepID2 respectively. Grayscale images of two modalities are identified by training the LeNet-5 model. BN and Dropout are adopted to optimize the above convolutional neural network (CNN) models. From Table 3 and Figure 2, we can get better recognition rates of each modality under different frequencies by using the proposed approach than by some other methods. The most remarkable features can be extracted by both the DCNN model we constructed and Gabor transform of images, which enhances the recognition rates.

3.2 Final System and the Computation of Accuracy

The problem of features fusion based on RGB-D images will be discussed in the following and the final recognition rates will be calculated. Five kinds of features (five frequencies) of each modality are extracted and are connected in series. The features extracting from two modalities are concatenated to a vector expression as the final features, and then SVM is chosen as a classifier to achieve

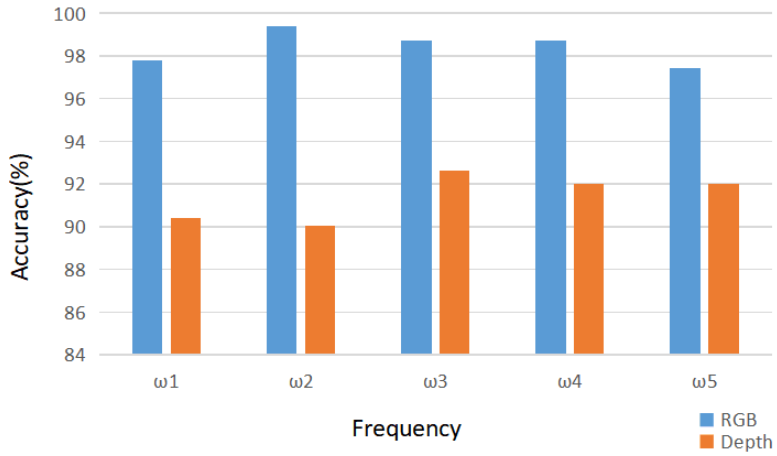


Figure 2: The recognition rates trained by the model in five frequencies

classification task.

We also compare the proposed method with RISE [8], LBP [1], LeNet-5 and DeepID2. For RISE algorithm, Viola-Jones detection is carried out at first, then the entropy maps of RGB images and Depth maps are computed respectively, and saliency maps of RGB images are calculated. The features are extracted from entropy and saliency maps by the histogram of oriented gradient (HOG), and then Random Decision Forest (RDF) is selected as a classifier to calculate recognition rates. LBP, just calculating the binary result between the center pixel and the neighbors, is used to extract features from three channels of images, and then SVM is chosen as a classifier to obtain the final result of object recognition. Both DeepID2 and LeNet-5 are utilized to extract features as extractors, features from two modalities are connected into a vector, and then SVM is chosen to achieve object recognition. The experimental results are shown in Table 4.

From Table 4, we can see that our algorithm achieves an excellent result. The identification rate using the proposed method is about 11% higher than that of the RISE algorithm, about 5% higher than that of the LBP algorithm. Since the DeepID2 and LeNet-5 cannot perform well for the cropped Depth maps by face detection, the recognition rates of two modalities images for classification task of EURECOM data set are low. The experimental results show that the proposed method performs well and is also competitive.

4 Conclusions

This paper proposes a new method, named Gabor-DCNN, applied for RGB-D face recognition. RGB-D can help solve fundamental problems due to its complementary nature of the Depth information and the visual information of the RGB. Gabor filters can magnify the changes of gray level and enhance the local features of some critical functional areas of the face such as eyes, nose, mouth, eyebrows, and so on, which are helpful to distinguish different face images. Gabor feature maps transformed by Gabor filters have excellent spatial locality and directional selectivity, which are robust to illumination and posture, and therefore have been successfully applied in face recognition. Convolutional neural network (CNN) can automatically achieve image classification and get higher accuracy than other traditional methods. The constructed DCNN model is utilized to extract features. The Gabor-DCNN, combining Gabor transform and the deep convolutional neural network can dramatically improve the classification accuracy. The experimental results on RGB-D data set validate the effectiveness of the proposed method. The proposed algorithm performs more excellently and obtains better results of object classification than some state-of-the-art.

Although we get better performance on face recognition of EURECOM data set by the proposed algorithm, the drawback of the Gabor-DCNN lies in the way of classification, which needs to extract features of two modalities RGB-D and be linked together, and then a classifier is chosen to achieve categorization. Namely, feature extractors and classifiers are separated in the proposed approach. In order to solve the problem, future work will focus on the design of the parallel neural network for face recognition, which can achieve object classification automated.

Acknowledgments

This work is supported by Natural Science Foundation of Guizhou, China under Contract LH[2014]7641, by the National Statistic Bureau of China under Contract 2016LY81. The authors thank Yu Feng and Hainan Wang for their generous assistance and valuable suggestions.

Table 4: The comparison of acccuracy in different methods on EURECOM data set

Method	The Proposed	DeepID2	LeNet-5	RISE	LBP
Accuracy	97.76	69.55	77.56	86.22	92.63

References

[1] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *European Conference on Computer Vision*, pp. 469–481, 2004.

[2] A. M. Basbrain, J. Q. Gan, and A. Clark, "Accuracy enhancement of the viola-jones algorithm for thermal face detection," in *Intelligent Computing Methodologies*, pp. 71–82, 2017.

[3] Z. Baochang, S. Shiguang, C. Xilin, and G. Wen, "Histogram of Gabor phase patterns (HGPP): A novel object representation approach for face recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 57–68, 2006.

[4] Y. L. Boureau, F. Bach, Y. Lecun, and J. Ponce, "Learning mid-level features for recognition," in *Computer Vision & Pattern Recognition*, 2010. (`https://www.di.ens.fr/willow/pdfs/cvpr10c.pdf`)

[5] L. Chengjun and W. Harry, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 11, no. 4, pp. 467–476, 2002.

[6] M. Chihaoui, A. Elkefi, W. Bellil, and C. B. Amar, "Implementation of skin color selection prior to gabor filter and neural network to reduce execution time of face detection," in *International Conference on Intelligent Systems Design & Applications*, 2016. DOI: 10.1109/ISDA.2015.7489251.

[7] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in Neural Information Processing Systems*, vol. 2, no. 4, pp. 396–404, 1990.

[8] G. Goswami, M. Vatsa, and R. Singh, "RGB-D face recognition with texture and attribute features," *IEEE Transactions on Information Forensics & Security*, vol. 9, no. 10, pp. 1629–1640, 2014.

[9] I. Y. H. Gu, D. P. Kumar, and Y. Yun, "Privacy-preserving fall detection in healthcare using shape and motion features from low-resolution RGB-D videos," in *International Conference Image Analysis & Recognition*, pp. 490–499, 2016.

[10] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," *International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2014.

[11] G. E. Hinton and N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on International Conference on Machine Learning*, 2015. (`https://arxiv.org/pdf/1502.03167.pdf`)

[13] Y. Ji, A. Yamashita, and H. Asama, "RGB-D SLAM using vanishing point and door plate information in corridor environment," *Intelligent Service Robotics*, vol. 8, no. 2, pp. 105–114, 2015.

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, and F. F. Li, "Large-scale video classification with convolutional neural networks," in *Computer Vision & Pattern Recognition*, 2014. (`https://static.googleusercontent.com/media/research.google.com/zh-TW//pubs/archive/42455.pdf`)

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012. (`https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`)

[16] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. V. D. Malsburg, R. P. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture. ieee transactions on computers, 42, 300-311," *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, 1993.

[17] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *IEEE International Conference on Robotics & Automation*, pp. 1817–1824, 2011.

[18] J. B. C. Neto and A. N. Marana, "Face recognition using 3DLBP method applied to depth maps obtained from kinect sensors," in *X Workshop de Visão Computacional*, 2014. (`file:///C:/Users/user/Downloads/0029.pdf`)

[19] S. Qian and D. Chen, "Discrete gabor transform," *IEEE Transactions on Signal Processing*, vol. 41, no. 7, pp. 2429–2438, 1993.

[20] S. T. H. Rizvi, G. Cabodi, P. Gusmao, and G. Francini, "Gabor filter based image representation for object classification," in *International Conference on Control*, 2016. DOI: 10.1109/CoDIT.2016.7593635.

[21] M. Rui, N. Kose, and J. L. Dugelay, "KinectFaceDB: A kinect database for face recognition," *IEEE Transactions on Systems Man & Cybernetics Systems*, vol. 44, no. 11, pp. 1534–1548, 2014.

[22] A. Schmidt, M. Fularz, M. Kraft, A. Kasiński, and M. Nowicki, "An indoor RGB-D dataset for the evaluation of robot navigation algorithms," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 321–329, 2013.

[23] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. DOI: 10.1109/CVPR.2015.7298655.

[24] H. Wang, B. Zhang, H. Zheng, Y. Cao, Z. Guo, and C. Qian, "The robust derivative code for object recognition," *Ksii Transactions on Internet & Information Systems*, vol. 11, no. 1, pp. 272–287, 2017.

[25] X. Wang, L. U. Youtao, S. Song, and X. Ping, "Face recognition based on Gabor wavelet transform and modular pca," *Computer Engineering & Applications*, vol. 48, no. 3, pp. 176–176, 2012.

[26] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. Mcdonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 598–626, 2015.

[27] T. Whelan, M. Kaess, J. J. Leonard, and J. Mcdonald, "Deformation-based loop closure for large scale dense RGB-D SLAM," in *IEEE / RSJ International Conference on Intelligent Robots & Systems*, 2013. `http://thomaswhelan.ie/Whelan13iros.pdf`

[28] A. Wilkowski, T. Kornuta, and W. Kasprzak, "Point-based object recognition in RGB-D images," *Intelligent Systems*, vol. 323, pp. 593–604, 2015.

[29] S. Xue, "Face database security information verification based on recognition technology," *International Journal of Network Security*, vol. 21, no. 4, pp. 601–606, 2019.

[30] H. Yao, C. Li, H. Dan, and W. Yu, "Gabor feature based convolutional neural network for object recognition in natural scene," in *International Conference on Information Science & Control Engineering*, 2016. DOI: 10.1109/ICISCE.2016.91.

[31] C. Ziyun, H. Jungong, L. Li, and S. Ling, "RGB-D datasets using microsoft kinect or similar sensors: A survey," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4313–4355, 2017.

Biography

Yuanyuan Xiao had received M.S. in School of Electronic information Engineering from Guizhou University in 2008; she is with the School of Computer Science and Technology from Guizhou University as a Ph.D. candidate. Her research interests include machines learning, deep learning and virtual reality.

Xiaoyao Xie is a professor with Key Laboratory Of Information and Computing Science of Guizhou Province, Guizhou Normal University. He is also a member of the China Computer Federation. His current research interests include computer network, information security, machines learning and pattern classification.

Classification of DoS Attacks in Wireless Sensor Network with Artificial Neural Network

Munawar Hussain, Jiadong Ren, and Awais Akram
(Corresponding author: Munawar Hussain)

Department of Information and Engineering, Yanshan University
438 Hebei Street West Section, Haigang, Qinhuangdao, Hebei, China
(Email: munawarjut@yahoo.com)
(Received June 12, 2019; Revised and Accepted Dec. 3, 2019; First Online Feb. 28, 2020)

Abstract

Due to deployment in sensitive military areas and other security applications, wireless networks are becoming a famous research spot in the field of computer science. To ratify the security and reliability in such kinds of application intrusion detection system can play an important role. There is a need of intrusion detection system, which has the capability to detect a large number of possible threats in wireless sensor networks. This article contains a customized dataset for wireless sensor network that can be categorized into four types of DoS attacks (Blackhole, Grayhole, Flooding & Scheduling attacks). For the experimental purpose, since it is highly used in WSN, Low Energy Aware Cluster Hierarchy (LEACH) protocol has been used in this research work. Using NS 2 network simulator to model a scheme which define to collect network traffic and create the dataset. Artificial Neural Network has been applied to train the dataset to classify it into different DoS attacks. Experimental work performed here gives high classification rate and accuracy for mentioning attacks with the help of proposed dataset. In future, suggested method can be useful for more attacks like Sybil/Wormhole presented in datalink layer as DoS attacks.

Keywords: Artificial Neural Network; DoS Attacks; LEACH Protocol; Machine Learning; Multilayer Perceptron (MLP); Wireless Sensor Network

1 Introduction

Wireless Sensor Network (WSN) is becoming more imperative research area due to its large employment in diverse kind of applications, for example health care, building monitoring, smart cities deployment and sensitive military areas *etc.* [14, 18]. The deployment of WSN nodes are usually done in wide areas to collect the data and to transmit the data in intelligent way to the base station [8]. The transmission of data in WSN is based on dedicated WSN protocols. Securing WSN is becoming main challenge because of limited resources of WSN like processing power, memory and battery backup [2]. Generally WSN nodes are deployed in an unprotected and hostile environment. This makes nature of WSN more vulnerable [13].

These limitations put constraint on existing security methods like cryptography, which are not always enough for this type of networks. WSN is always surrounded of attacks due to its deployment in open areas with limited resources. In WSN during frequent transmission of the packets, attackers can compromise with sensor node and may cause drop of the message, inject and alteration of the message, interrupt the integrity of the data as well as waste the energy resources. Denial of Services (DoS) attack is most common and dangerous attack for WSN. This attack may cause to interrupt and hang the WSN services [6].

Due to weak security prevention arrangements an Intrusion Detection System it is required to detect the unwanted attacks (known and unknown). IDS is complex for WSN due to its small size nodes and limited hardware. We have also found out that there is no specific dataset which have normal and abnormal traffic logs for these kind of experimental work [17].

In the light of above mentioned challenges IDS designing need two main characteristics. First IDS must be highly accurate for any attack detection secondly it must have low weight with minimum cost or overhead. In this paper we have built a specific dataset for WSN which have attack model which can be able to categorize DoS attack. LEACH protocol used to combat the WSNs? energy consumption challenges.

The remaining paper is distributed as mentioned. Section II included the idea of LEACH protocol and related work. Section III shows the dataset description and creation. Section IV mentioned experimental results and discussion. Last section summarized the conclusion and future work.

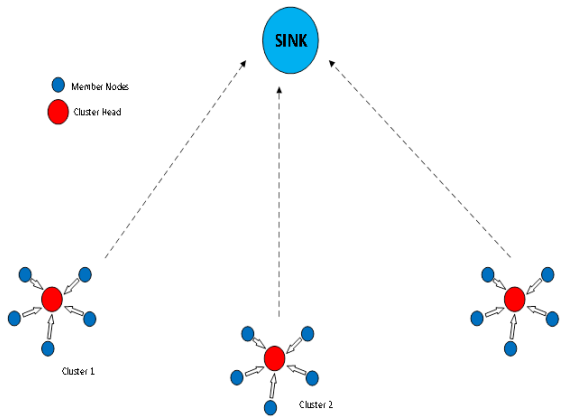


Figure 1: Nodes structure in LEACH routing protocol

2 Idea of LEACH Protocol & Related Work

The main Idea of the proposed LEACH Protocol was to reduce the energy consumption and to enhance the lifetime of sensor network, it?s a protocol based on clustering with self-organizing capability [9]. In LEACH usually BS (Base Station) is located far away from the nodes and nodes are similar to each other with limited space and power. Figure 1 shows the arrangement of nodes in LEACH which contain two steps on every round. Round one is step round where cluster are made and second round known as steady round where sensed data transferred to sink node. LEACH is still being mentioned and explored by many researchers. Authors in [4, 20] reviewed many clustering routing methods depending upon LEACH protocol for wireless sensor networks with detailed discussions and judgements. Another author in [12] presented improved versions of LEACH protocol. Authors have compared and discussed few features of LEACH protocol. In [3] authors have developed mechanisms to reduce the amount of Redundant data transmission in WSN to reduce the overall energy of the network. In [15] author evaluated and developed new clustering based principles for heterogeneous WSN based on LEACH protocol. Author [19] introduced LEACH Inner Clustering Election method depends upon LEACH. To improve clustering LEACH-ICE select new cluster head (CH) into the cluster which energy of current CH become lower as mentioned threshold. In the research [7] author planned an algorithm to select CH selection for wireless sensor network by managing distance among CH uniform distribution of CHs performed. In experimental area new method showed good performance for consumption of energy and life time of network.

3 Attack Model

The main aim of DoS is to create commotion in services by trying to limit the access to a service or machine instead of destabilizing the service itself. DoS attacks include

Black holes are places in a network when the network's incoming or outgoing traffic starts getting dropped or discarded silently or without getting the sources informed. The source are not even notified that their packets are not reaching the intended destinations. The only way to detect the Black holes in a network topology is by monitoring the lost traffic [16]. Algorithm 1 used for this type of attack. Flooding as its name states is made by intentionally sending a very high volume of traffic to a network node such that it can not allow or examine the permitted network traffic. For example in a TCP connection establishment during the three way handshaking an attacker can repeatedly, may be huge amount of times, initiate the handshake SYN and not respond afterwards. Since the listening server waits for the third handshake step that is SYN-ACK every time and dedicating small amount of resources to keep that session open. Eventually the server cannot handle any more resulting in no legitimate users to be able to log on [16]. Algorithm 3 used for this kind of attack.

Grayhole attacks are special case of DoS attacks and are kind of selective forwarding attacks. The nodes selectively discards or drops few, not all, packets which were supposed to be forwarded along the path [11]. Algorithm 2 used for this type of attack.

Wormhole attacks rre again a special type of DoS attack that misleads routing operations. The malicious nodes create tunnels through which the packets are replayed to malicious nodes hence corrupting the network routing procedures [10]. Algorithm 4 used for this kind of attack.

In recent days many researchers are trying to find the solutions for DoS attacks in WSNs but mainly they have found one or two or partial solutions with high energy consumption [21, 22]. So a suitable system should be developed to mention and classify different attitude of DoS of attacks.

IDS has become essential part of security mechanism while IDS application for WSN is still challengeable. There are two basic part of IDS feature extraction and modelling algorithm. Feature extraction id responsible to describe measured attributes which associates to IDS function. Modelling attributes responsible for accuracy and proficiency in IDS. Few important IDS are described as Monitoring, Analysis, Detection and three component described as action in IDS including Logging, Alarming and Prevention as shown in Figure 2.

4 Dataset Description & Creation

The main aim in this paper to create a dataset to classify the DoS attacks. Many works has been done to improve the IDS strategies. Important factor involved in this process is the data used for testing and training of the detection model. The better the data quality the better

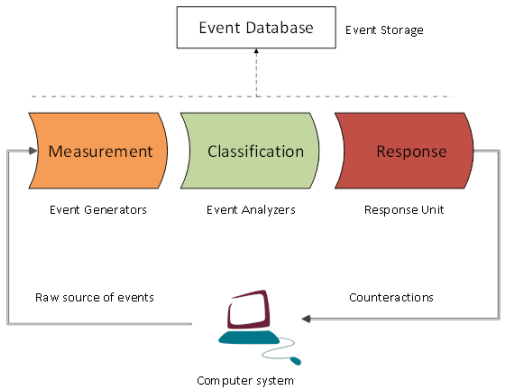


Figure 2: Nodes structure in LEACH routing protocol

the offline intrusion detection system can be analysed. In the research of IDS techniques the KDD data set (Knowledge Discovery and Data Mining) is a well-known benchmark [1]. Mostly KDD dataset has been constructed for LAN environments only, these dataset were not specific for wireless environment although many researchers used it for intrusion detection purposes. There is a need to develop a dataset for WSN to observe normal and abnormal behaviour. In order to create the dataset for WSN we needed the dataset related to wireless environment which can help in detection classification of DoS attacks. In this study distribution of the traffic among the nodes and every node is responsible in monitoring. Dataset attributes are described in Table 1.

5 Mathematical Model for LEACH

For WSN dataset construction a mathematical model developed to ensure the correctness of dataset, the term used in this model are following Table 2.

This model used to analysis in different phases including,

Advertisement phase: This step used to calculate the number of advertisements send by CHs and receive by CMs.

Cluster setup: This step responsible to calculated the join requests message send by nodes and received by related CHs.

Transmission of data: This step responsible to calculate the sensed data delivered to BS.

6 Experimental Results and Discussion

For collection of data NS2 simulation software used with following parameters settings as shown in Table 2. In this paper LEACH protocol is used to create the dataset,

Algorithm 1 Blackhole attack algorithm (Algorithm 3):

```
Begin
if rSNi ¡ TSNi then
   SNi=CH
else
   SNi=CM.
   if CHj, J = NC then
      X CMs join CHj
      CHj create TDNA schedule
   end if
   if CHj=MN  then
      Execute attack by dropping packets
   else
      Send data to BS

   end if
end if
End
```

Algorithm 2 Grayhole attack algorithm (Algorithm 3):

```
Begin
if rSNi ¡ TSNi then
   SNi = CH
else
   SNi = CM
   if CHj, J NC then
      X CMs join CHj
      CHj create TDNA schedule
   end if
   if CHj=MN  then
      CHj = Adv_CH (with high broadcasting)
   else
      CHj = Adv_CH (normal broadcasting)

   end if
end if
if xCMs join CHj  then
   CHj = Adv_CH (with high broadcasting)
else
   CHj create TDMA schedule
   xCMs send data to related CHj in associated
   TDMA time slice
end if
End
```

Dataset contain 374668 records which can be categorized into four kinds of DoS attack (Blackhole, Grayhole, flooding and Scheduling). Multilayer Perceptron (MLP) Artificial neural network classifier configuration used in this paper. MLP is highly famous ANN variation which enables to configure more than one layer ANN, and useful to build complex relation among input and output values.

Due to different performance confusion metrics we have used following performance matrix. These performance matrix are True Positive Rate (TPR), True Negative (TN), False Positive (FP), False Negative (FN) and Pre-

Table 1: Dataset attributes

Node ID	Matchless ID for differentiate the sensor node
Time	Simulation time
RSSI	Received signal strength indication
Space to CH	Distance between nodes
Max distance to CH	Maximum space between Cluster Head
Average distance to CH	Average of distance between cluster head
Current energy	Recent energy status of node
Energy consumption	Volume of energy to be used in last round
ADV CH send	Broadcast of messages by CH for sending
ADV CH receives	Received messages by CH
Join REQ send	Joining messages to CH by nodes
Join REQ receive	Received messages by CH for joining nodes
ADV SCH send	TDMA (Time Division Multiple Access) messages to nodes
ADV SCH receives	TDMA messages received from CH
Rank	Node order in TDMA
Data sent	Nodes data packages sent to CH
Send Code	Cluster code
Rank	Node order in TDMA
Data received	Data packages received from CH
Attack Type	Classification of attacks

Table 2: Description of mathematical model for LEACH

Term	Description
N	Number of nodes in network
Si	Node i
NC	CH numbers
CM	Cluster members
ADV CH Sent	Advertisement messages by CH
ADV-CH-RCVD	Advertisement received by nodes
Join Req Sent	Join requests by nodes
Join Req Rcvd	Join requests received by CH
TDMA Sent	TDMA routine send by CH
TDMA RCVD	TDMA routine received by nodes
NO PKT	Number of data packets received by CH

Algorithm 3 Scheduling attack algorithm (Algorithm 3):

```
Begin
if rSNi ¡ TSNi then
    SNi = CH
else
    SNi = CM
    if CHj, J NC then
        X CMs join CHj
        CHj create TDNA schedule
    end if
    if CHj process attack by structuring TDMA routine, give same time slice to all
    node for send the data then
        CHj = Adv_CH (with high broadcasting)
    else
        CHj structure normal TDMA routine

    end if
end if
if xCMs join CHj then
    CHj = Adv_CH (with high broadcasting)
else
    CHj xCMs send data to CHj in related TDMA time slice
    CHj send data to BS
end if
End
```

cision. Formulas 01 to 03 taken from [5]. Few MLP ANN structure developed in this paper with one, two and three hidden layers. Dataset was separated into training 60 % and testing set 40% as showing in Table 4.

We use two kind of cross validation method to train the dataset. Cross validation is used to evaluate the model. The problem with residual evaluations is that they do not give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on "new" data. This is the basic idea for a whole class of model evaluation methods called cross validation. Dataset divided 70% training set and 30 testing sets for experiment purpose as mentioned below table.

Table 3: Dataset attributes

Parameters setting	Numbers values
Data packets size	500 bytes
Network length	100 x 100 meters
Packet headers size	25 bytes
Protocol	LEACH
Experimental time	1 hr
Cluster numbers	Five
Transmission limitation	200 meters

Holdout cross validation: This is simple type of cross validation by setting the data into train and test. By using approximator function only for train set. All data is randomly divided into same equal size data sets. The advantage of holdout method is mostly common to residual method and take short time for calculation.

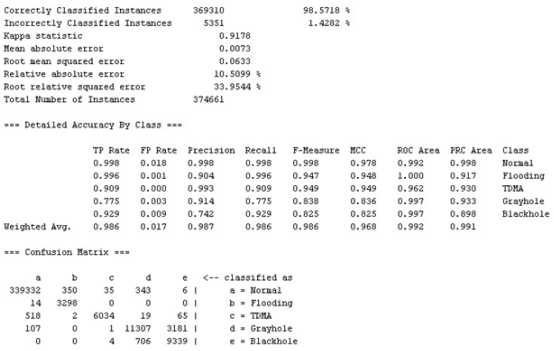
K-fold cross validation: This is an improved form for previous method where dataset divvied into given number for example 10 and holdout method recurrent k number of time. Advantage of this method is calculated variance is reduced as k number is increased.

Table 4: Dataset was separated into training 60 % and testing set 40%

Attack Types	Train Data	Test Data
Scheduling	3982	2656
Flooding	1988	1324
Grayhole	8758	5838
Blackhole	6029	4020
Normal	204039	136027

In this article dataset constructed using LEACH protocol in WSN. Dataset have 374668 records by representing four kinds of DoS attacks. Multilayer Artificial neural network machine learning (MLP) method is used to train dataset by having more than one hidden layer. Due to different performance confusion metrics we have used following performance matrix. These performance matrix are True Positive Rate (TPR), True Negative (TN), False Positive (FP), False Negative (FN) and Precision. Formulas 01 to 03 taken from [5]. Following Tables 4, 5, 6 and 7 describes the output summery of holdout cross validation method for TPR, FPR, FNR, TNR under hidden Layers I, II and III.

Classification is best for all type of attacks but not satisfactory for Grayhole and Scheduling attacks. Figure 3 for FPR having lower ratio and the good performance under hidden layer 1. Figures 4 and 5, shows good performance except for FNR. Figures 6 and 7 shows the simulation process by using single hidden layer. Figures 8, 9 and 10 shows the output performance with up to three



```
Correctly Classified Instances      369310               98.5718 %
Incorrectly Classified Instances      5351                1.4282 %
Kappa statistic                          0.9178
Mean absolute error                      0.0073
Root mean squared error                  0.0633
Relative absolute error                 10.5099 %
Root relative squared error             33.9544 %
Total Number of Instances           374661

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   MCC      ROC Area  PRC Area  Class
                 0.998     0.018     0.998     0.998     0.998     0.978     0.992     0.998     Normal
                 0.996     0.001     0.904     0.996     0.947     0.948     1.000     0.917     Flooding
                 0.909     0.000     0.993     0.909     0.949     0.949     0.962     0.930     TDMA
                 0.775     0.003     0.914     0.775     0.838     0.836     0.997     0.933     Grayhole
                 0.929     0.009     0.742     0.929     0.825     0.825     0.997     0.898     Blackhole
Weighted Avg.    0.986     0.017     0.987     0.986     0.986     0.968     0.992     0.991

=== Confusion Matrix ===

      a      b      c      d      e   <-- classified as
 339332    350     35    343      6 |      a = Normal
     14   3298      0      0      0 |      b = Flooding
    518      2   6034     19     65 |      c = TDMA
    107      0      1  11307   3181 |      d = Grayhole
      0      0      4    706   9339 |      e = Blackhole
```

Figure 3: Simulation output with single layer

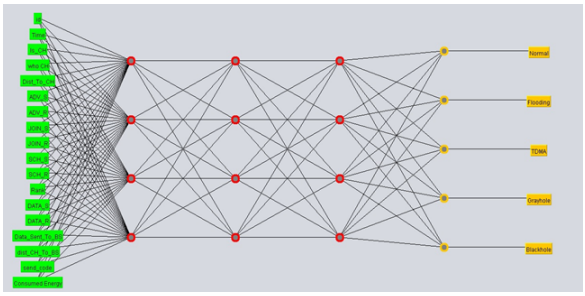


Figure 4: GUI simulation process with three hidden layers

hidden layer by using 10 cross validation method. Its observes that by using cross validation method CV method batter with single hidden layer for classification of normal and blackhole attacks.

So it is noted that cross validation architecture highly batter in the classification for denial of service attack in WSN. By obtaining the result with collected WSN dataset and applying MLP ANN is the possibility of highly accurate output for classification of denial of service attacks.

7 Conclusions

The idea of this paper to create the structure of Intrusion Detection System which have ability to work against DoS

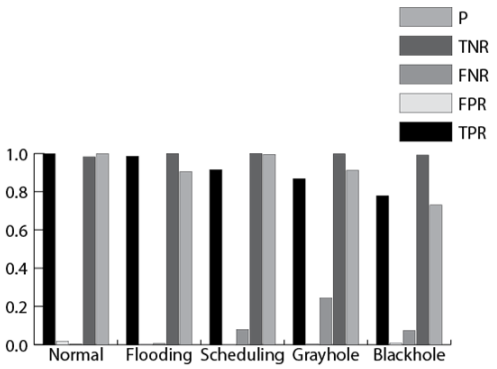


Figure 5: 10 cross validation method with one hidden layer

Table 5: True positive ratio summery with holdout cross validation

Hidden Layer	Normal	Flooding	Scheduling	Grayhole	Blackhole
I	0.99	1	0.98	0.92	0.34
II	0.99	1	0.96	0.71	0.81
III	0.98	0.95	0.97	0.57	0.98

Table 6: False positive ratio summery with holdout cross validation

Hidden Layer	Normal	Flooding	Scheduling	Grayhole	Blackhole
I	0.008	0.003	0	0.006	0.611
II	0.004	0.002	0	0.003	0.004
III	0.316	0.003	0.001	0.001	0.616

Table 7: False negative ratio summery with holdout cross validation

Hidden Layer	Normal	Flooding	Scheduling	Grayhole	Blackhole
I	0.004	0	0.014	0.679	0.657
II	0.004	0	0.16	0.286	0.182
III	0.005	0.011	0.027	0.424	0.011

Table 8: True negative ratio summery with holdout cross validation

Hidden Layer	Normal	Flooding	Scheduling	Grayhole	Blackhole
I	0.997	0.997	1	0.997	0.997
II	0.992	0.996	1	0.995	0.987
III	0.988	0.996	0.990	0.998	0.983

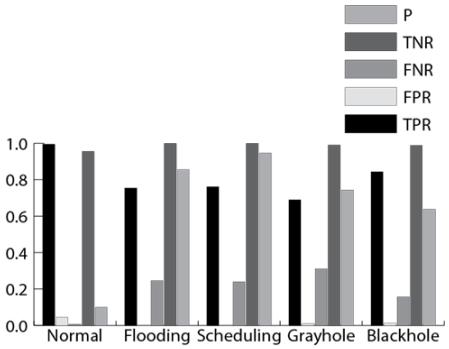


Figure 6: 10 cross validation method with three hidden layer

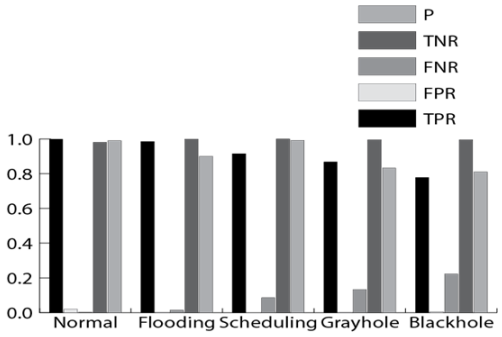


Figure 7: 10 cross validation method with two hidden layer

attacks with in affordable cost. The conclusion can be summarized with considering the results which are successfully classified as the DoS attacks with high detection rate. The future work will be extended with results having more hidden layers as well as able to classify more network attacks like wormhole attack or Sybil.

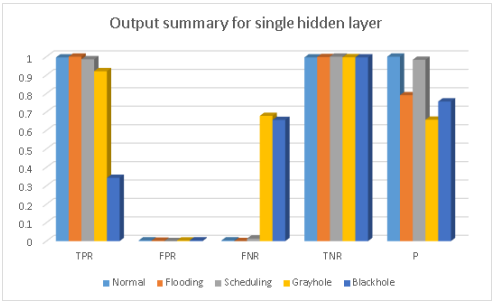


Figure 8: Holdout method Results with single hidden layer

References

[1] P. Aggarwal, S. K. Sharma, "Analysis of KDD dataset attributes - Class wise for intrusion detection," *Pro-*

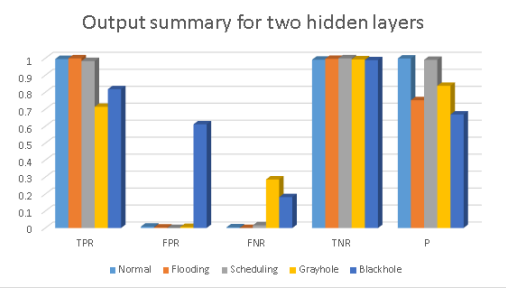


Figure 9: Holdout method Results with two hidden layer

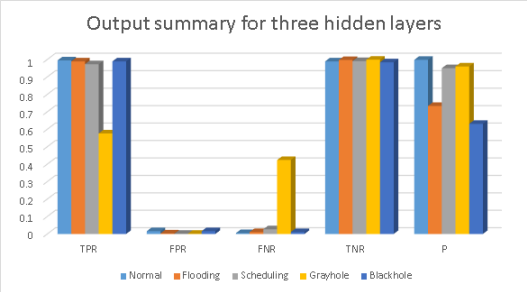


Figure 10: Holdout method Results with three hidden layer

cedia Computer Science, vol. 57, pp. 842-851, 2015.

[2] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 266–282, 2014.

[3] R. N. Enam, N. Ismat, F. Farooq, "Connectivity and coverage based grid-cluster size calculation in wireless sensor networks," *Wireless Personal Communications*, vol. 95, pp. 429-443, July 2017.

[4] R. N. Enam, M. Tahir, R. Qureshi, "A survey of energy conservation mechanisms for dynamic cluster based wireless sensor networks," *Mehran University Research Journal of Engineering & Technology*, vol. 37, no. 2, pp. 279, 2018.

[5] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision 111*, no. 1, pp. 98-136, 2015.

[6] N. Farooq, I. Zahoor, S. Mandal, and T. Gulzar, "Systematic analysis of DoS attacks in wireless sensor networks with wormhole injection," *International Journal of Information and Computation Technology*, vol. 4, no. 2, pp. 173–182, 2014.

[7] A. Garofalo, C. D. Sarno, and V. Formicola, "Enhancing intrusion detection in wireless sensor networks through decision trees," in *Dependable Computing*, pp. 1–15, 2013.

[8] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557– 3564, 2010.

[9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd IEEE Annual Hawaii International Conference on System Sciences*, pp. 1–10, 2000.

[10] S. K. Jangir, N. Hemrajani, "A comprehensive review on detection of wormhole attack in MANET," in *International Conference on ICT in Business Industry & Government (ICTBIG'16)*, pp. 1-8, 2016.

[11] R. Kaur, P. Singh, "Review of black hole and grey hole attack," *The International Journal of Multimedia & Its Applications*, vol. 6, pp. 35-45, 2014.

[12] D. Kumar, "Performance analysis of energy efficient clustering protocols for maximizing lifetime of wireless sensor networks," *IET Wireless Sensor Systems*, vol. 4, no. 1, pp. 9–16, 2014.

[13] M. Kumar, K. Dutta, I. Chopra, "Impact of wormhole attack on data aggregation in hieracrchical WSN," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 70–77, 2014.

[14] N. Marriwala and P. Rathee, "An approach to increase the wireless sensor network lifetime," in *Proceedings of the World Congress on Information and Communication Technologies (WICT'12)*, pp. 495–499, 2012.

[15] Y. M. Miao, "Cluster-head election algorithm for wireless sensor networks based on LEACH protocol," *Applied Mechanics and Materials*, vol. 738-739, pp. 19–22, 2015.

[16] S. Patil, S. Chaudhari, "DoS attack prevention technique in wireless sensor networks," *Procedia Computer Science*, vol. 79, pp. 715-721, 2016.

[17] M. A. Rassam, M. A. Maarof, and A. Zainal, "A survey of intrusion detection schemes in wireless sensor networks," *American Journal of Applied Sciences*, vol. 9, no. 10, pp. 1636–1652, 2012.

[18] R. Singh and M. S. Manu, "An energy efficient grid based static node deployment strategy for wireless sensor networks," *International Journal of Electronics and Information Engineering*, vol. 7, no. 1, pp. 32-40, 2017.

[19] S. Taneja, "An energy efficient approach using load distribution through LEACH-TLCH protocol," *Journal of Network Communications and Emerging Technologies (JNCET'15)*, vol. 5, no. 3, pp. 20–23, 2015.

[20] S. Tyagi and N. Kumar, "A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 623–645, 2013.

[21] S. S. Wang, K. Q. Yan, S. C. Wang, and C. W. Liu, "An integrated intrusion detection system for cluster-based wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15234–15243, 2011

[22] J. Xu, J. Wang, S. Xie, W. Chen, and J. U. Kim, "Study on intrusion detection policy for wireless sensor networks," *International Journal of Security and Its Applications*, vol. 7, no. 1, pp. 1–6, 2013.

Biography

Munawar Hussain biography. Munawar Hussain is pursuing his PhD Degree from Yanshan university, China. He completed his master degree from University of Education Lahore, Pakistan. His interested area is complex network, Network security, Data mining and big data.

Jiadong Ren biography. Jiadong Ren is a professor at Yanshan University, Qinhuangdao, China. He is also a supervisor of doctoral and master students. His research area is data mining, network security and software security.

Awais Akram biography. Awais Akram currently pursuing his PhD Degree from Yanshan University China, He completed his Master degree from Islamia University Bahawalpur. His research interested area is Complex network security and data mining.

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.