IJNS
International Journal of Network Security

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

Vol. 17, No. 3 (May 2015)

INTERNATIONAL JOURNAL OF NETWORK SECURITY

PUBLISHING OFFICE

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

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

PUBLISHER: Candy C. H. Lin

International Journal of Network Security

Vol. 17, No. 3 (May 1, 2015)

A False Data Report Filtering Scheme in Wireless Sensor Networks: A Survey

Tung-Huang Feng[1], Wei Teng Li[2], and Min-Shiang Hwang[1,3]
(Corresponding author: Min-Shiang Hwang)

Department of Computer Science & Infoarmation Engineering, Asia University[1]
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung, Taiwan, R.O.C.
Department of Management Information System, National Chung Hsing University[2]
Department of Medical Research, China Medical University Hospital, China Medical University[3]
(Email: mshwang@asia.edu.tw)
(Invited Jan. 20, 2014; revised and accepted May 4, 2014)

Abstract

Wireless sensor networks (WSNs) have been used in many areas and contain a large number of tiny sensor nodes in the environment. An intruder can easily capture a large number of sensor nodes because these sensor nodes are deployed in the unattended environment. Using these compromised nodes an intruder can inject the false data report to the network. The false reports make the server misjudge and not respond the real situation immediately. To prevent false data report injection attacks, false data report filtering schemes have been proposed in recent years. Existing false data report filtering schemes are classified into two categories: symmetric key based schemes and asymmetric key based schemes. In this survey paper, we survey the previous researches of false data report filtering schemes based on a symmetric key in wireless sensor networks. Our contributions are to analyze the previous researches regarding their advantage and disadvantage.

Keywords: False data report filtering, message authentication code, wireless sensor network

1 Introduction

Wireless sensor networks (WSNs) have been used in many areas and contain a large number of tiny sensor nodes in the environment. These sensor nodes are used to sense and collect data in the unattended environment [5]. Then sensor nodes send these data to the base station. The base station analyzes these data and make a decision. Due to the cost, sensor nodes lack a tamper-resistance, which consists of sensing unit, processing unit, power unit, communication and transceiver [1, 16]. Therefore, vulnerable sensor nodes are always the attack target of an intruder [6, 7, 8, 9, 12].

An intruder can easily capture a large number of sensor nodes because these sensor nodes are deployed in the unattended environment [3, 17] and get the secret values in the sensor nodes such as public key, private key and session key. Using these compromised nodes, an intruder can inject the false data report to the network. When an event happens, the nodes close to the event will generate a report and use its symmetric key to make a message authentication code (MAC), which is then sent to the base station (BS). If an intruder deploys his malicious sensor nodes, he can use these compromised sensors to generate the fake report and inject the false report to the network. The false reports make the server misjudge and not respond the real situation immediately.

To prevent false data report injection attacks, false data report filtering schemes have been proposed in recent year. When an event occurs, the nodes closest to the event call detecting nodes to generate a report and use its symmetric key to make a message authentication code (MAC) and send to the base station (BS). Then BS will check the report whether it would contain enough number of $MACs$ or not. Filtering false data report schemes have to ensure the network not to be disturbed by the false data reports so to make server not respond the real report immediately. When detecting false data reports in the network, the network has to drop the fake report instantly.

Existing false data report filtering schemes are classified into two categories: *symmetric key based schemes* [18, 20, 24, 25, 26] and *asymmetric key based schemes* [2, 14, 19, 22, 23]. Asymmetric key based schemes such as RSA cryptographic algorithm are not suitable to be used in resource-constrained wireless sensor network environments, because the computation is too intensive to afford for wireless sensor networks. Most of existing related works are based on a symmetric key. In this paper, we survey the previous researches of false data report filtering schemes based on symmetric key in wireless sensor networks. Our contributions are to analyze the previous

researches regarding their advantage and disadvantage. Finally, we also mention the future work about the false data report filtering scheme and give the conclusions.

The rest of paper is organized as follows: Section 2, we classify the basic requirements of security and efficiency used to analyze previous researches. Section 3, we discus the existing schemes of false data report filtering in detail. Section 4, we analyze previous researches and demonstrate their pros and cons. Finally, we summarize and discuss the future work of false data report filtering schemes in wireless sensor networks in Section 5.

2 Basic Requirements and Evaluation Metrics

According to [4, 11, 15, 21] surveyed papers, these papers provide the basic requirements of security and efficiency. In our paper, we also give the basic requirements of filtering false data report schemes. We classified these requirements into two categories: security metrics and efficiency metrics. Then we use these requirements to analyze the existing schemes in Section 4.

2.1 Security Metrics

Filtering false data report schemes have to ensure the network not to be disturbed by the false data reports which can make servers not respond the real report immediately. When detecting false data reports in the network, the network has to drop the fake report instantly. In filtering false data report schemes, each node has to use its symmetric key to generate a message authentication code when an event happens. An intruder can capture a large number of sensor nodes in the network and extract their symmetric keys. An intruder can use these symmetric keys to generate fake message authentication codes and send these codes to the server. Therefore, four requirements are classified for filtering false data report schemes: false data report revocation, forward and backward secrecy, collusion resistance and resilience.

1) False data report revocation: If an intruder use the malicious sensor nodes to inject the false report to try disturbing the network so to make servers not respond the real report immediately or eavesdrop the communication of other sensors. Then secure filtering false data report schemes have to filter the false data report and drop them immediately not to affect the real function of wireless sensor networks.

2) Forward and backward secrecy: Forward secrecy indicates that even if an intruder steals the old keys from legitimate sensor nodes, he cannot use these old keys to continue decrypting current messages. Backward secrecy is used to protect a compromised node from knowing the current new keys and going back to decipher previous messages received by an intruder. To prevent node capture attacks, forward and backward secrecies are very important in wireless sensor networks.

3) Collusion resistance: An intruder might compromise a number of sensor nodes in the network and extract their secret keys. And these secret keys are used to inject the false data report to make a server break down. A secure scheme can prevent an intruder from collaborative false data injection attacks.

4) Resilience: Resilience is used to describe the strength and toughness of wireless sensor networks. An intruder might compromise a number of sensor nodes in the wireless sensor network. If the resilience of the filtering false data report scheme is weak, few false data reports will lead to the whole network breakdown. On the other hand if resilience is high, the network can tolerate a number of false data reports in the network and cannot affect the function of wireless sensor networks.

2.2 Efficiency Metrics

Under the limited resources of wireless sensor network environment, energy saving is a very important thing. Both memory and energy consumptions should be as low as possible. All overheads must be overall reduced, such as computational complexity, storage and operations.

1) Memory: Due to the cost, sensor nodes generally have not sufficient storage. Sensor nodes usually store its identity, node's public, private key, session key and its neighboring members. Because of the limited memory, the storage capacity of sensor nodes has to be as small as possible.

2) Energy: Saving energy is another challenge in wireless sensor networks. Data transmission and data reception will lead to a lot of energy consumption [10]. However, in false data report filtering schemes, it involves lots of data transmission and data reception. Hence, designing a lightweight scheme is beyond doubt.

3 False Data Report Filtering Scheme

In this section, we discuss the related works of false data report filtering schemes in detail. We review each of these schemes and analyze their contribution. We also highlight the pros and cons of the existing literatures. In false data report filtering schemes, when an event happens, the nodes closest to the event will generate a report and use its symmetric key to make a message authentication code (MAC) which are sent to the base station (BS). Then BS will check whether the report contains enough numbers of $MACs$ or not. We will analyze the difference of each scheme and explain the limitations of

various schemes. The notation for false data report filtering schemes are listed in Table 1.

Table 1: Notations for false data report filtering scheme

BS	Base station
CH	Cluster head
ID_α	Node α's identity
$H(.)$	A one-way hash function
$MAC_k(M)$	MAC of message M using a symmetric key k
L_E	Location of event
t	Time of detection
E	Event information

3.1 SEF Scheme

In SEF protocol proposed by Ye et al. [24], there is a global key pool containing N keys. The global key pool is divided into n distinct partitions and each of n distinct partitions consists of m keys in these partitions. Before deployment, each node needs to choose k keys from one partitions. For example, as shown in Figure 1, the global key pool with $n = 8$ partitions and three nodes need to choose $k = 3$ keys from one partitions. When an event appears, the nodes closest to the event collaboratively prepare a legitimate report. The form of report is $\{L_E, t, E\}$. A detecting node α selects one of its k keys K_{α_i} and makes an MAC which contains $M_i = MAC_{K_{\alpha_i}}(L_E||t||E)$. Then the node α sends $\{i, M_i\}$ to the CH, where i is the key index. The CH collects all $\{i, M_i\}$ from the neighboring nodes closest to the event, and the final report sent to the BS looks like $\{L_E, t, E, i_1, M_{i1}, i_2, M_{i2} \cdots, i_T, M_{iT}\}$.

In the filtering phase, when the BS receives the final report, it first checks whether the report carries T key indices and T $MACs$. Here, T means a secret threshold value that is just known by the BS. If the final report contains less than T threshold value, the report will be dropped immediately. Otherwise, the BS verifies the correctness of each MAC in the final report. If the BS finds out an error in MAC, the whole packet will be dropped. For example, as shown in Figure 2, the detecting nodes are Nodes 1, $\cdots$, 5. If $T = 5$, the final report will be accepted and then used to verify the correctness of each MAC in the final report. If $T = 10$, the final report in Figure 2 will be dropped, because the final report contains less than $T = 10$. Although SEF provides a simple key assignment method for filtering false data report, SEF falls into collaborative false data injection attacks [20]. An intruder can easily capture T nodes and extract their symmetric keys. Then it uses these keys to forge T $MACs$ in the report which is sent to the BS. BS will be cheated by an intruder. Because the final report contains T key indices and T $MACs$. The fake report may be sent to many hops before it is detected. This leads to wasting too much energy in the limited resources of the wireless sensor network environment.

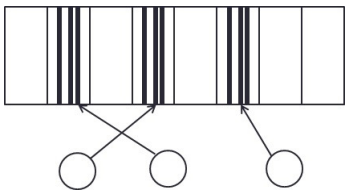


Figure 1: Global key pool

3.2 GRSEF Scheme

A group-based filtering scheme, called GRSEF [25], was proposed by Yu et al. in 2009. In GRSEF, after deployment, all the nodes in the network are divided into T groups. This allows the high probability of the different groups may cover any position. Before deployment, every node is preloaded with a global master key K_g, and K_g is used to compute its group master key $K_{gi} = H_{K_g}(i)$, where i means group number. Then K_{gi} is combined with the multiple axes-based method to derive its authentication keys $K_{p_u^i}$. When an event happens, the detecting nodes generate an MAC using their authentication keys $MAC_{K_{p_u^i}}(E)$. Then it sends $\{i, MAC_{K_{p_u^i}}(E)\}$ to CH. The CH collects all $\{i, MAC_{K_{p_u^i}}(E)\}$ from the neighboring nodes closest to the event, and the final report is sent to BS looks like $R = \{i, MAC_{K_{p_u^i}}(E) | 1 \le i \le T\}$. GRSEF is shown in Figure 3.

In the filtering phase, when a node in group g_i receives the event report, it can use the event location to derive partition-binding keys to verify the correctness of the MAC from its group. If the forwarding nodes find out an error in MAC or the number of $MACs$ is less than T, the whole packet will be dropped. When the packet is sent to the BS, BS can derive all the keys by using K_g and verify all $MACs$ in the report. However, computing multiple axes-based key and maintaining these keys will cost a large of communication cost.

3.3 DEFS Scheme

A dynamic en-route filtering scheme named DEFS [26] was proposed by Yu et al. in 2010. In DEFS, first all nodes in the networks are divided into several groups and each group has a cluster head (CH). Each node is preloaded with a distinct initial seed key k_m, and every node has a different initial seed key k_m. The k_m can use a hash function to generate a sequence of authentication keys $k_1, k_2, \cdots, k_m$. The CH establishes several different paths to the BS, and Hill Climbing algorithm is used to disseminate the first authentication key to the forwarding nodes on multiple paths. When an event happens, node α_i generates a report $r(\alpha_i) = \{E, \alpha, j_i, MAC_{k_{j_i}^{\alpha_i}}(E)\}$, where j_i is the index of α_i's current authentication key. The CH

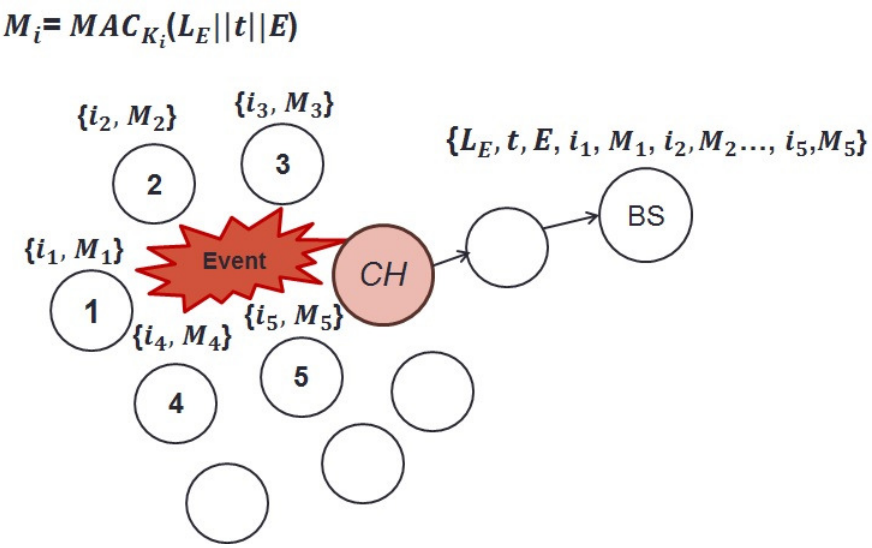


Figure 2: SEF scheme

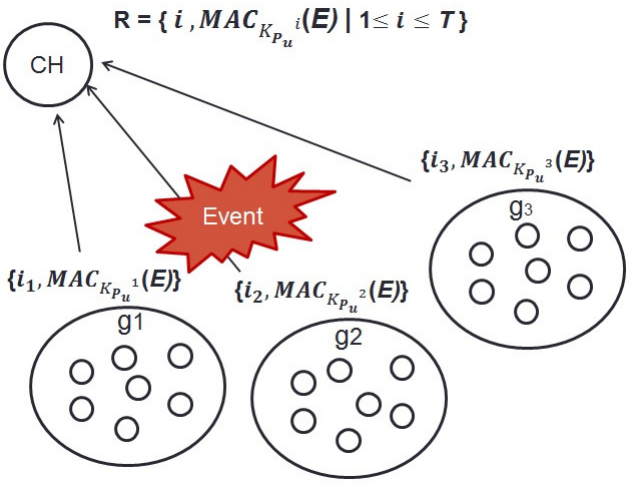


Figure 3: GRSEF scheme

collects all the detecting nodes' reports and aggregates the reports $R = \{r(\alpha_{i1}), \cdots, r(\alpha_{iT})\}$.

In the filtering phase, first, each forwarding node receives the reports from its upstream node. Second, it will wait a confirmation OK message from its upstream nodes and then sends the report to the next forwarding node. If it does not receive a confirmation OK message, it will drop the reports. Third, it receives a message $K(T)$, which contains the disclosed authentication keys, and then uses the disclose keys to check the $MACs$ in the report. If the $MACs$ are correct, it sends a confirmation OK message to the next forwarding node. However, DEFS seems not suitable for real WSNs environment. In the key distributed phase, DEFS consumes a lot of energy and WSNs cannot afford the multi-path based forwarding.

3.4 DSF Scheme

A double key-sharing based false data filtering scheme, called DSF [18], was proposed by Sun et al. in 2013. In DSF, nodes are divided into several clusters, and each cluster has a cluster head (CH). Before deployment, nodes are preloaded with their unique ID. In DSF, there are two-types of keys to ensure the security of wireless sensor networks. One of the keys is called R-type keys. Like SEF, there is a global key pool containing N keys. The global key pool is divided into n distinct partitions and each of n distinct partitions consists of m keys in these partitions. Another key is called A-type key. The A-type keys are pairwise keys which are established with its associated nodes. Establishment of the associated nodes is based on [13] algorithm. Each node in the same cluster has its own associated node and establishes a pairwise key with its associated node. When an event happens, node α will use its R-type and A-type key to generate two $MACs$: MAC_R, MAC_A. Then α sends $\{ID_\alpha, MAC_R, MAC_A\}$to CH. The final report is like $\{R_1, \cdots, R_t; MAC_{R_1}, \cdots, MAC_{R_T}; A_1, \cdots, A_t; MAC_{A_1}, \cdots, MAC_{A_T}\}$, where R,A are R-type key and A-type key from the detecting nodes;MAC_R, MAC_A use R-type and A-type key to generate the $MACs$.

In the filtering phase, the forwarding nodes receive the final report, it first checks whether the final report contains enough T threshold value R-type, A-type keys and MAC_R, MAC_A or not. If not, the final report will be dropped. Then the forwarding nodes check whether the T R-type keys come from different partitions; if there are two keys from the same partition, the report will be dropped. Final, if the forwarding nodes contain one of the R-type keys or the pairwise keys in the final report, they would use them to generate the $MACs$ to see whether the corresponding $MACs$ are correct or not. If all the checking is correct, it will send the report to the next forward-

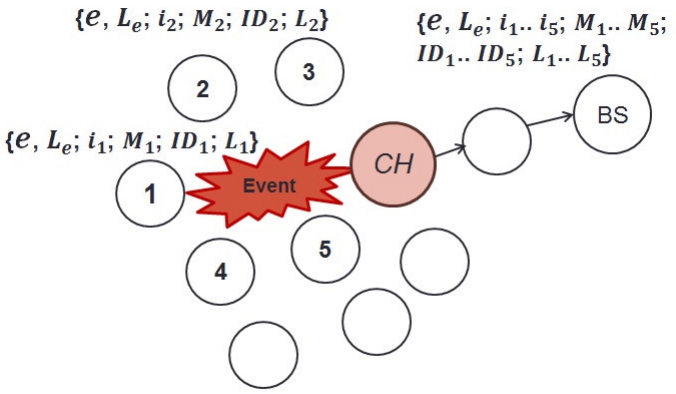


Figure 4: GFFS scheme

ing node. However, DSF uses two types of keys to achieve double-check of the final report, it cannot defend the collaborative false data injection attacks. When an intruder compromises a number of sensor nodes in the wireless sensor network, he not only knows the R-type keys, but also knows A-type keys. The intruder can launch collaborative false data injection attacks and disturb the wireless sensor networks.

3.5 GFFS Scheme

A geographical information based false data filtering schemes, named GFFS [20], was proposed by Wang et al. in 2014. This paper proposed GFFS scheme which can defend collaborative false data injection attacks. Most of the existing related works only consider whether the final report contains enough T threshold value $MACs$ or not. They did not consider the correctness of the sensor nodes. If an intruder compromises numbers of sensor nodes in the network, he can generate the final report which contains T threshold value $MACs$ in it. And the BS is cheated by enough T threshold value $MACs$ report. In GFFS predeployment phase, like SEF, there is a global key pool containing N keys. The global key pool is divided into n distinct partitions and each of n distinct partitions consists of m keys in these partitions. Each node randomly chooses one partition and then selects k ($k < m$) keys in the selected partition. After deployment, each node has its location L_i. When an event occurs, as shown in Figure 4, the detecting nodes generate a detecting report which contains$\{e, L_e; i; MAC; ID_i; L_i\}$ where e is an event information;L_E is the event location; i is the key index; ID is the node's identity and L_i is node's location. Then it sends to the CH and CH and collects all the detecting report $\{e, L_e; i_1, \cdots, i_T; MAC_{i1}, \cdots, MAC_{iT}; ID_1, \cdots, ID_T; L_{ID_1}, \cdots, L_{ID_T}\}$.

In the filtering phase, after the forwarding nodes receive the report, it first checks whether the report contains enough T threshold key indices and $MACs$. Then it checks if the key indices belong to different partitions. Next, it checks whether the report contains enough T ID_T and node's location (L_{ID_T}). If not, the report will be dropped. Each node has its sensing radius r_s. The forwarding nodes calculate the distance between the detecting node and the event location. If L_i and L_e are in the sensing radius r_s, the L_i is legitimate. Otherwise, the L_i is compromised, and the report will be dropped. Then if the forwarding node possess any key indices in the report, it generates the MAC by using its key and compares the correctness of the corresponding MAC in the report. If the $MACs$ are correct or the forwarding nodes do not have any keys in the report, the nodes forward the report to the next hop. However, GFFS can defend the collaborative false data injection attacks from an intruder by compromising numbers of sensor nodes from different geographical locations, but it cannot defend the intruder who compromises sensor nodes within the sensing radius r_s. If an intruder captures sensor nodes within r_s and compromises enough T threshold sensor nodes, GFFS will fail to defend the collaborative false data injection attacks. For example, as shown in Figure 5, If $T = 5$, an intruder can capture Nodes 1 to 5 and compromise them. The report still can pass the detection.

4 Discussions

In this section, we classify the pros and cons of the above existing literatures in Table 2. We also use the basic requirements mentioned in Section 2 to analyze the security and efficiency of above related works. In Section 2, we have summarized the security metrics and evaluation metrics. We use these requirements to discuss these false data report filtering schemes. According the result, it shows the fact that there still are lots of challenges needed to overcome.

Table 2: Summary of scheme

Scheme	Advantage	Disadvantage
SEF [24]	1.Independent data dissemination 2.Simple protocol	1.T-threshold limitation problem 2.Higher energy consumption 3.Collaborative false data injection attack
GRSEF [25]	1.More resilience 2.Improve T-threshold problem 3.Location aware manner	1.Higher communication cost 2.Hard to maintenance
DEFS [26]	1.More resilience 2.Dynamic authentication key 3.Multipath routing	1.Complex key distribution 2.Higher energy consumption
DSF [18]	1.More resilience 2.Double key sharing	1.Complex pairwise key establishing 2.Partial defend collusion attack
GFFS [20]	1.More secured 2.Geographical information based	1.Expensive positioning device

Table 3: Summary of scheme security

Scheme	Node revocation	Collusion resistance	Resilience	Lightweight
SEF [24]	Yes	No	No	Low
GRSEF [25]	Yes	Yes	Partial	Medium
DEFS [26]	Yes	Yes	Partial	Medium
DSF [18]	Yes	Yes	Partial	Medium
GFFS [20]	Yes	No	Partial	High

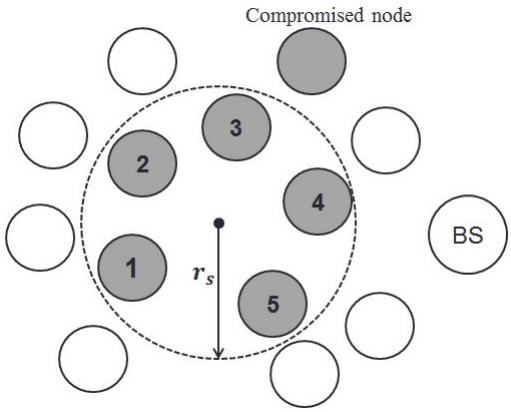


Figure 5: Collaborative false data injection attacks

4.1 Security and Performance Analysis

In Section 2, we have summarized the security metrics and evaluation metrics. We use these requirements to discuss these false data report filtering schemes and summarize them in Table 3. Resilience is used to describe the strength and toughness of wireless sensor networks. "High" indicates the false data report can not disturb the network and would filter immediately. "Low" means the false data report cannot efficiently filter and lead to networks paralysis. We list the storage overhead in Table 4.

Table 4: Summary of scheme costs

Scheme	Storage
SEF [24]	0.4
GRSEF [25]	2.4
DEFS [26]	3.5
DSF [18]	1.2
GFFS [20]	0.5

5 Future Research and Conclusions

In false data report filtering schemes, false data report filtering is based on setting an appropriate secret T threshold value. According to the above related works, a legitimate report must contains enough T $MACs$ in it. If the $MACs$ in the report is less than T threshold value, BS will regard this report illegitimate and drop it immediately. The future works of false data report filtering schemes have to be designed to defend collaborative false data injection attacks. If an intruder captures enough numbers of sensor nodes in the network, he can use these compromised nodes to launch the false data injection attacks. Defending collaborative false data injection attacks in false data report filtering scheme is still an important task. For implementing in resource-constrained wireless sensor networks environments, the design of the false data report filtering scheme must be as lightweight as possible.

Defending false data infection is an important task in wireless sensor networks. In our survey, we focus on false data report injection attacks and survey the existing literatures of these topics. We analyze the advantage and disadvantage of these related works and discuss them in detail. We also give the requirement of the security and efficiency to discuss the related works in detail. According to the literatures, the future works of false data report filtering schemes have to be designed to defend collaborative false data injection attacks and implemented for a real wireless sensor networks environment.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant NSC 102-2811-E-468 -001, 103-2622-E-468 -001 -CC2, and 103-2622-H-468 -001 -CC2. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] E. Ayday, F. Delgosha, and F. Fekri, "Location-aware security services for wireless sensor networks using network coding," in *IEEE Conference on Computer Communications*, pp. 1226–1234, 2007.

[3] B. S. Babu, N. Jayashree, and P. Venkataram, "Performance analysis of steiner tree-based decentralization mechanism (STDM) for privacy protection in wireless sensor networks," *International Journal of Network Security*, vol. 15, no. 5, pp. 331–340, 2013.

[4] X. He, M. Niedermeier, and H. de Meer, "Dynamic key management in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 611–622, 2013.

[5] V. Katiyar, N. Chand, and N. Chand, "Recent advances and future trends in wireless sensor networks," *International Journal of Applied Engineering Research*, vol. 1, no. 3, pp. 330–342, 2010.

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

[7] C. T. Li, M. S. Hwang, and Y. P. Chu, "Improving the security of a secure anonymous routing protocol with authenticated key exchange for ad hoc networks," *International Journal of Computer Systems Science and Engineering*, vol. 23, no. 3, pp. 227–234, 2008.

[8] C. T. Li, M. S. Hwang, and Y. P. Chu, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks," *Computer Communications*, vol. 31, no. 12, pp. 2803–2814, 2008.

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

[10] J. Li and J. Li, "Data sampling control and compression in sensor networks," *Mobile Ad-hoc and Sensor Networks*, vol. 3794, pp. 42–51, 2005.

[11] W. T. Li, T. H. Feng, and M. S. Hwang, "Distributed detecting node replication attacks in wireless sensor networks: A survey," *International Journal of Network Security*, vol. 16, no. 5, pp. 323–330, 2014.

[12] W. T. Li, C. H. Ling, and M. S. Hwang, "Group rekeying in wireless sensor networks: A survey," *International Journal of Network Security*, vol. 16, no. 6, pp. 401–410, 2014.

[13] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *Proceedings of the 1st ACM workshop on Security in Ad Hoc and Sensor Networks*, pp. 72–82, 2003.

[14] K. Naresh, K. P. PrDadeep, and K. S. Sathish, "An active en-route filtering scheme for information reporting in wireless sensor networks," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 4, pp. 1812–1819, 2012.

[15] M. A. Jr. Simplicio, P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, "A survey on key management mechanisms for distributed wireless sensor networks," *Computer Networks*, vol. 54, no. 15, pp. 2591–2612, 2010.

[16] A. Singla and R. Sachdeva, "Review on security issues and attacks in wireless sensor networks," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 4, pp. 529–534, 2013.

[17] H. S. Soliman and M. Omari, "Application of synchronous dynamic encryption system (SDES) in wireless sensor networks," *International Journal of Network Security*, vol. 3, no. 2, pp. 160–171, 2006.
[18] Q. Sun and M. Wu, "A double key-sharing based false data filtering scheme in wireless sensor networks," *Journal of Computers*, vol. 8, no. 2, pp. 388–398, 2013.
[19] H. Wang and Q. Li, "A public-key based false data filtering scheme in sensor networks," in *Proceedings of the International Conference on Wireless Algorithms, Systems and Applications*, pp. 129–138, 2007.
[20] J. Wanga, Z. Liu, S. Zhang, and X. Zhang, "Defending collaborative false data injection attacks in wireless sensor networks," *Information Sciences*, vol. 254, no. 1, pp. 39–53, 2014.
[21] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Computer Communications*, vol. 30, no. 11, pp. 2314–2341, 2007.
[22] F. Yang, X. H. Zhou, and Q. Y. Zhang, "Multi-dimensional resilient statistical en-route filtering in wireless sensor networks," *Lecture Notes in Computer Science*, pp. 130–139, 2010.
[23] H. Yang and S. Lu, "Commutative cipher based en-route filtering in wireless sensor networks," in *Vehicular Technology Conference*, pp. 1223–1227, 2004.
[24] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 839–850, 2005.
[25] L. Yu and J. Li, "Grouping-based resilient statistical en-route filtering for sensor networks," in *Proceedings of 28th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1782–1790, 2009.
[26] Z. Yu and Y. Guan, "A dynamic en-route filtering scheme for data reporting in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 150–163, 2010.

Tung-Huang Feng received his M.S. in Information Management from Chao-Yang University of Technology, Taichung, Taiwan, ROC, in 2002. He is currently pursuing the Ph.D. degree from Computer Science & Information Engineering Department of Asia University, Wufeng, Taiwan. His research interests include information security, cloud computing, and Sensor Networks.

Wei-Teng Li received his B. M. in Management Information Systems from National Chung Hsing University, Taichung, Taiwan, ROC, in 2012. He is currently pursuing the M.S. degree with the Department of Management Information Systems from National Chung Hsing University. His research interests include information security, wireless sensor network, and cryptography.

Min-Shiang Hwang received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China (ROC), in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2002. He was a professor and chairman of the Graduate Institute of Networking and Communications, CYUT, during 2002-2003. He is currently a professor of the department of Management Information System, National Chung Hsing University, Taiwan, ROC. He obtained 1997, 1998, 1999, 2000, and 2001 Outstanding Research Awards of the National Science Council of the Republic of China. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang had published 170+ articles on the above research fields in international journals.

A Secure Anonymous Authentication Scheme for Wireless Communications Using Smart Cards

Yanrong Lu[1,2], Xiaobo Wu[3], and Xiaodong Yang[4]
(Corresponding author: Yanrong Lu)
Information Security Center, State Key Laboratory of Networking and Switching Technology[1]
Beijing University of Posts and Telecommunications, Beijing 100876, China
National Engineering Laboratory for Disaster Backup and Recovery[2]
Beijing University of Posts and Telecommunications, Beijing 100876, China
School of Software Engineering, Yantai Vocational College, Shandong Yantai 264670, China[3]
College of Computer Science and Engineering, Northwest Normal University, Gansu Lanzhou 730070, China[4]
(Email: luyanrong1985@163.com)
(Received Jan. 10, 2015; revised and accepted Feb. 22 & Mar. 23, 2015)

Abstract

Wireless communications have become one of the most key parts in our everyday life, which give us facility on work and life but they still bring a great security risk. A crucial problem with wireless communications is to ensure the security of communication and prevent the privacy of communication entities revealing. Authentication is becoming an important issue when a mobile user (MU) wants to access services provided by the home agent (HA) in a visited foreign agent (FA). Recently, Kuo et al. proposed an authentication scheme and claimed that the proposed scheme was secure against different kinds of attacks. In this paper, we show that Kuo et al.'s scheme fails to resist insider and verifier attacks while it does not provide local verification. In addition, password change phase of Kuo et al.'s scheme also has a loophole. To remedy these shortcomings, an improved anonymous authentication scheme for wireless communications is proposed which is immune to various known types of attacks. Finally, in comparison with other existing schemes regarding security properties and performance, we show that our scheme has various kinds of security properties and is suitable for practical applications in wireless networks.

Keywords: Authentication, key establishment, smart cards, wireless communications

1 Introduction

Wireless communications technologies [7] have experienced swift development to satisfy the increasing demands of travelling farther and faster in practical applications. Mobile devices within range of a wireless network can transmit data anywhere and anytime. One consequence of this communication is that a malicious adversary can eavesdrop the transmitted messages so that he can impersonate as a legal participant to enjoy services. This is naturally bringing the important issues of the protection of privacy among the users in a public channel [20]. Authentication scheme [2, 3, 4, 5, 6, 12, 14, 15, 19, 21] is a mechanism to authenticate a remote MU over an open network. If the MU roams into a foreign network, they must be authenticated by the FA with the help of the user's HA. Therefore, it is necessary and meaningful for constructing a secure authentication scheme in wireless communication to ensure communications security.

In recent years, many authentication schemes have been proposed for wireless communications. Zhu et al. [23] presented an anonymous authentication scheme for wireless networks. However, Lee et al. [13] found that Zhu et al.'s scheme could not provide mutual authentication and then proposed an enhanced authentication scheme to conquer the weakness. Later, Wu et al. [22] and Chang et al. [6] respectively showed that both Zhu et al. and Lee et al.'s scheme also did not achieve anonymity. And then they respectively proposed their modified authentication scheme to eliminate the flaw. However, Mun et al. [1] pointed out that Wu et al.'s scheme failed to provide perfect forward secrecy. To remedy the deficiency, Mun et al. proposed an improved one. Unfortunately, Kim et al. [18] found that Mun et al.'s scheme was also vulnerable to replay and man-in-the middle attacks. Recently, Kuo et al. [8] proposed an Elliptic Curve Cryptography (ECC) [10] based authentication scheme and claimed that their scheme can achieve many security goals. However, by carefully analysis, we show that Kuo et al.'s scheme still has one or more weaknesses.

In this paper, we analyze Kuo et al.'s scheme and find that it is susceptible to resist insider and verifier attacks while it does not provide local verification. And their

password change phase has a loophole. To overcome these security weaknesses, an enhanced authentication scheme is proposed by us. We demonstrate that our scheme satisfies several security properties. In addition, performance and security comparison show that our scheme is also efficient and secure compared with other related schemes.

The layout of the paper is organized as follows. Section 2 introduces some difficult problems about the ECC. The review and security analysis of Kuo et al.'s scheme are shown in Section 3 and Section 4, respectively. Sections 5 and 6 show our proposed scheme and analyze its security. Section 7 depicts the performance and functionality comparison among the proposed scheme and other related ones. Section 8 is a brief conclusion.

2 Preliminaries

In this section, we first introduce some difficult problems about the ECC [17]. Next, notations adopted throughout this paper are listed in Table 1.

The hard problems:

1) The computational discrete logarithm (CDL) problem is given points A, B, where $B = mA$, decide $m \in \mathbb{F}_q^*$.

2) The computational Diffie-Hellman (CDH) problem is given points mP, nP, compute mnP over $E_p(a, b)$.

Table 1: Notations

MU/FA	Mobile user / Foreign agent
HA	Home agent
ID_A/PW_A	Identity / password of an entity A
$h(\cdot)$	Hash function
p_A	Secret key selected by A
$\oplus$	Exclusive-or operation
$\|\|$	Concatenation operation
N_A/r_A	A random number selected by an entity
$P.x$	x-axis value of the point on the ECC
p_{HA-MU_i}	Secret key of HA for MU
t_A	Timestamp generated by an entity A
$E_k(\cdot)$	Symmetric encryption using key k
$D_k(\cdot)$	Symmetric decryption using key k

3 Review of Kuo et al.'s Scheme

In this section, we recall the Kuo et al.'s scheme. Their scheme consists of four phases: registration, authentication with key agreement is shown in Figure 1, update session key and password change phases.

Registration Phase:

1) MU selects his identity ID_{MU}, and secret key p_{MU}. Then, he computes $PW_{MU} = h(ID_{MU}||p_{MU})$ and delivers $\{ID_{MU},\ PW_{MU}\}$ to HA through a secure channel.

2) On receiving the message, HA examines whether ID_{MU} already exists. If it does not exist, HA selects a random number N_{MU_i} and then computes $U = h(p_{HA-MU_i}||N_{MU_i})$, $W_i = PW_{MU} \oplus N_{MU_i}$, $V_i = N_{MU_i} \oplus p_{HA-MU_i}$. Then, HA stores $\{U,\ PW_{MU},\ p_{HA-MU_i}\}$ into its database and issues smart card including the information $\{ID_{HA},\ W_i,\ V_i,\ h(.)\}$ to MU through a secure channel.

Authentication with Key Agreement Phase:

1) MU inserts his smart card *thesmartcard* into a smart card reader and then keys his ID_{MU} and PW_{MU}. Then, the smart card generates $N_{MU_{i+1}}$ and computes $N_{MU_i} = PW_{MU} \oplus W_i$, $p_{HA-MU_i} = N_{MU_i} \oplus V_i$, $S_1 = h(p_{HA-MU_i}||N_{MU_i})$, $S_2 = PW_{MU} \oplus N_{MU_{i+1}}$, $S_3 = h(N_{MU_{i+1}}||ID_{FA})$ and $S_4 = h(PW_{MU}||h(p_{HA-MU_i}||N_{MU_{i+1}}))$. Finally, MU stores $N_{MU_{i+1}}$ and sends $\{ID_{HA},\ S_1,\ S_2,\ S_3,\ S_4\}$ to FA.

2) FA selects a random number a and computes aP. Then, FA stores $\{ID_{HA},\ aP\}$ and delivers $\{ID_{FA},\ S_1,\ S_2,\ S_3,\ S_4,\ aP\}$ to HA.

3) HA computes $N_{MU_{i+1}} = S_2 \oplus PW_{MU}$ and verifies $S_3' = h(N_{MU_{i+1}}||ID_{FA}) \stackrel{?}{=} S_3$ and $S_4' = h(PW_{MU} \oplus h(p_{HA-MU_i}||N_{MU_{i+1}})) \stackrel{?}{=} S_4$. If they are equal, HA computes $S_5 = h(PW_{MU}||N_{MU_{i+1}})$, $S_6 = h(ID_{FA}||ID_{HA}||S_5)$ and $S_7 = h(aP.x||S_5)$ and updates S_1 to $h(p_{HA-MU_i}||N_{MU_{i+1}})$. Finally, HA sends the message $\{ID_{FA},\ S_6,\ S_7,\ aP\}$ to FA.

4) FA checks whether the received ID_{HA} exists. If it exists, FA immediately submits $\{ID_{FA},\ S_6,\ S_7,\ aP\}$ to MU.

5) MU first checks

$$\begin{aligned} S_6' &= h(ID_{FA}||ID_{HA}||h(PW_{MU}||N_{MU_{i+1}})) \\ &\stackrel{?}{=} S_6, \\ S_7' &= h(aP.x||h(PW_{MU}||N_{MU_{i+1}})) \\ &\stackrel{?}{=} S_7. \end{aligned}$$

If they are consistent, MU chooses a random number b and computes bP, $K_{MF} = h(abP.x)$, $C_{MF} = h(K_{MF}||bP.x)$. Then, MU stores $\{W_i,\ V_i,\ aP\}$ and delivers $\{bP,\ C_{MF}\}$ to FA.

6) FA checks $C_{MF}' = h(K_{MF}||bP.x) \stackrel{?}{=} C_{MF}$. If they are equal, FA believes MU and stores $\{C_{MF},\ aP\}$ for session key update.

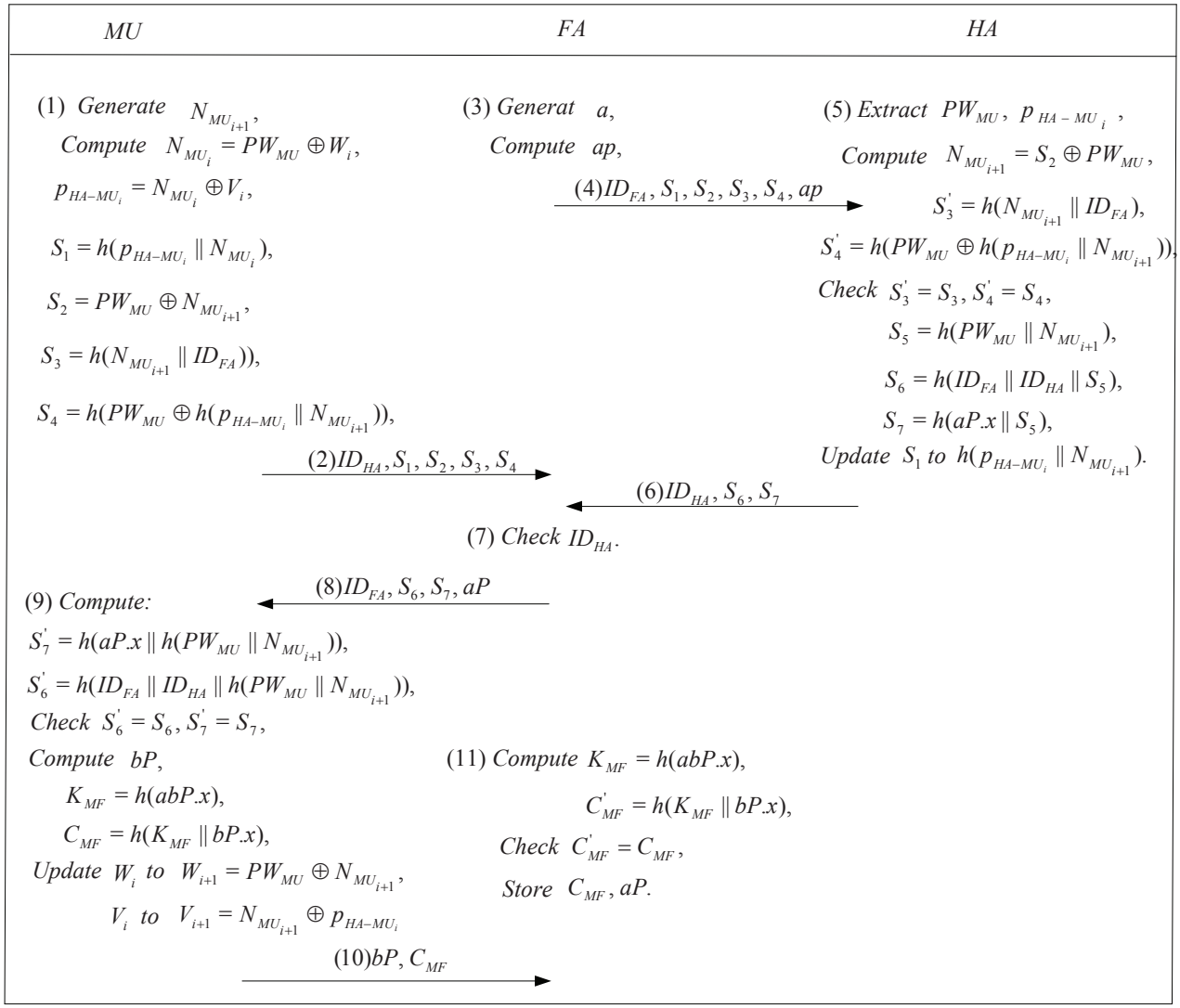


Figure 1: Authentication with key agreement of Kuo et al.' scheme

Update Session Key Phase:

1) MU chooses a random number b_i and computes b_iP. Then, MU transmits $\{b_iP,\ C_{MF_i}\}$ to FA.
2) FA first checks whether C_{MF_i} exists in its database. If it exists, FA generates a random number a_i and calculates a_iP, $K_{MF_{i+1}} = h(a_ib_iP.x)$, $C_{MF_{i+1}} = h(K_{MF_{i+1}}||b_iP.x)$, $h_1 = h(C_{MF_{i+1}}||a_{i-1}P.x)$. Finally, FA stores $\{C_{MF_{i+1}},\ a_iP\}$ and sends $\{a_iP,\ h_1\}$ to MU.
3) MU directly calculates $K_{MF_{i+1}} = h(a_ib_iP.x)$, $C_{MF_{i+1}} = h(K_{MF_{i+1}}||b_iP.x)$, $h_1' = h(C_{MF_{i+1}}||a_{i-1}P.x)$ and checks $h_1' \stackrel{?}{=} h_1$. If they match, MU authenticates the session key $K_{MF_{i+1}}$ and restores $\{C_{MF_{i+1}},\ a_iP\}$ into the device.

Password Change Phase:

1) MU chooses a random number p_{MU}^{new} and computes $PW_{MU}^{new} = h(ID_{MU}||p_{MU}^{new})$, $U = h(p_{HA-M_{U_i}}||N_{MU_i})$, $h_1 = (PW_{MU} \oplus PW_{MU}^{new})$ and $h_2 = h(PW_{MU}^{new}||p_{HA-M_{U_i}})$. Then, MU delivers $\{U,\ h_1,\ h_2\}$ to HA.
2) HA checks whether $h_2' = h((PW_{MU} \oplus h_1)||p_{HA-M_{U_i}}) \stackrel{?}{=} h_2$. If it holds, HA computes $h_3 = h(PW_{MU}||p_{HA-M_{U_i}})$ and updates PW_{MU} to PW_{MU}^{new} and sends h_3 to MU.
3) MU first checks $h_3' = h(PW_{MU}||p_{HA-M_{U_i}}) \stackrel{?}{=} h_3$. If it is equal, MU replaces W_i with $PW_{MU}^{new} \oplus N_{MU_i}$.

4 Cryptanalysis of Kuo et al.'s Scheme

Kuo et al. claimed their scheme can withstand various attacks. In this section, we demonstrate that their scheme is unable to protect against insider and verifier attacks while it does not provide local verification. And their password

change phase has a loophole. The following attacks are based on the assumptions that a malicious adversary $\mathcal{A}$ has completely monitor over the communication channel in authentication and establish the session key phase. So $\mathcal{A}$ can eavesdrop, modify, insert, or delete any message transmitted via public channel [11].

Insider Attack:

In the registration phase, MU directly transmits plaintext PW_{MU} to HA. Though HA is assumed to be trustworthy, there is a possibility that HA being an insider adversary $\mathcal{A}$ can impersonate MU after obtaining ID_{MU} and PW_{MU}. Therefore, Kuo et al.'s scheme cannot withstand insider attack.

Verifier Attack:

No matter which phases in the Kuo et al.'s scheme, FA and HA are in need of verification table to judge received data. For example, in the registration phase, HA needs to store MU's password PW_{MU} and each session random number p_{HA-MU_i} generated by HA. In the authentication and establishment of the session key phase, FA needs to store $\{ID_{HA}, C_{MF}, aP\}$. In the update session key phase, FA needs to store $\{C_{MF_{i+1}}, a_iP\}$ each update request. In the update password phase, HA needs to store the new password PW_{MU}^{new} each update request. If $\mathcal{A}$ steals the information $\{PW_{MU}, p_{HA-MU_i}\}$ and owns his smart card, he can use the stolen verifiers to impersonate as a legal participant of the scheme.

Unfriendly in Password Update Phase:

In order to change MU' password, MU must interact with HA to finish his will. Only MU is authenticated by HA and affirmed that the password update request is receivable, he can continue to change the password. We consider it is an inconvenient process for a user since he cannot update his password by himself without communicating with HA.

No Local Verification:

In authentication phase of Kuo et al.'s scheme, MU directly sends the login message to FA. Obviously, the smart card does not check whether the entered information is correct. Therefore, even if MU inputs the wrong information by mistake or $\mathcal{A}$ submits an forged message, the authentication scheme still continues without defected. This apparently leads to have extra communication and computational costs.

5 The Proposed Scheme

In this section, we present an enhanced ECC based anonymous authentication scheme. Our scheme also contains four phases: registration, authentication with key agreement is shown in Figure 2, session key change and the password change phases.

Registration Phase:

1) MU chooses his user name ID_{MU}, password PW_{MU}, and a random number r_{MU}^1. After that, he computes $C_{MU}^1 = h(PW_{MU}||r_{MU}^1)$ and submits $\{ID_{MU}, C_{MU}^1\}$ to HA over a secure channel.
2) On receiving the message, HA computes $n_{HA}^1 = C_{MU}^1 p_{HA}P$, $C_{HA}^1 = E_{p_{HA}}C_{MU}^1$. Then, he issues a smart card including $\{ID_{HA}, n_{HA}^1, C_{HA}^1\}$ and submits it to MU via a secure channel.
3) MU selects a random number r_{MU}^2 and computes $n_{MU}^1 = r_{MU}^2 n_{HA}^1$, $C_{MU}^2 = h(ID_{MU}||C_{MU}^1)$. Then, MU replaces n_{HA}^1 with n_{MU}^1 and adds C_{MU}^2 into the smart card. So the smart card becomes to contain the information $\{ID_{HA}, n_{MU}^1, r_{MU}^1, C_{HA}^1, C_{MU}^2\}$.

Authentication with Key Agreement Phase:

1) MU inserts his smart card into a card reader and enters his identity ID_{MU} and password PW_{MU}. Then the smart card checks $\overline{C_{MU}^2} = h(ID_{MU}||h(PW_{MU}||r_{MU}^1)) \stackrel{?}{=} C_{MU}^2$. If they are equal, the smart card computes $n_{MU}^2 = r_{MU}^2 P$, $n_{MU}^3 = C_{MU}^1 n_{MU}^2$, $C_{MU}^3 = E_{n_{MU}^3}(n_{MU}^1||C_{MU}^1||C_{HA}^1)$. Then, MU submits $\{ID_{HA}, C_{MU}^3, n_{MU}^1, t_{MU}, C_{HA}^1, n_{MU}^2\}$ to FA.
2) After receiving the message, FA checks whether t_{MU} is valid. If it is valid, FA delivers $\{ID_{FA}, C_{MU}^3, n_{MU}^1, t_{MU}\}$ to HA.
3) HA first computes $n_{MU}^3 = p_{HA}^{-1} n_{MU}^1$ by using his own secret key p_{HA}. It then decrypts C_{MU}^3 to reveal n_{MU}^1, C_{HA}^1 and C_{MU}^1. Next, HA decrypts the message C_{HA}^1 and compares the value of the message C_{MU}^1 in C_{HA}^1 with that of the C_{MU}^1 in C_{MU}^3. If they are equivalent, HA generates a random number r_{HA}^1 and computes $C_{HA}^2 = h(C_{MU}^1||n_{MU}^1||ID_{HA}||ID_{FA})$, $C_{HA}^3 = E_{k_{FH}}(ID_{HA}||C_{HA}^2||t_{HA})$, where k_{FH} is the shared secret key between HA and FA. HA sends the message $\{ID_{HA}, C_{HA}^3, t_{HA}\}$ to FA.
4) After receiving the message, FA decrypts C_{HA}^3 to reveal $\{ID_{HA}, C_{HA}^2, t_{HA}\}$ and checks whether ID_{HA} and t_{HA} are consistent with the received message. If they are consistent, FA selects a random number r_{FA}^1 and computes $n_{FA}^1 = r_{FA}^1 P$, $n_{MF} = r_{FA}^1 n_{MU}^2$ and the session key $SK_{MF} = h(n_{MF}||t_{MU})$. Finally, FA sends the message $\{ID_{FA}, C_{HA}^2, n_{FA}^1\}$ to MU.
5) Upon receiving the message, MU checks whether $\overline{C_{HA}^2} \stackrel{?}{=} C_{HA}^2$. If they are equal, MU computes $n_{MF} = r_{MU}^2 n_{FA}^1$ and establishes the common session key $SK_{MF} = h(n_{MF}||t_{MU})$.

Session Key Change Phase:

If MU wants to renew session key, it needs to agree on

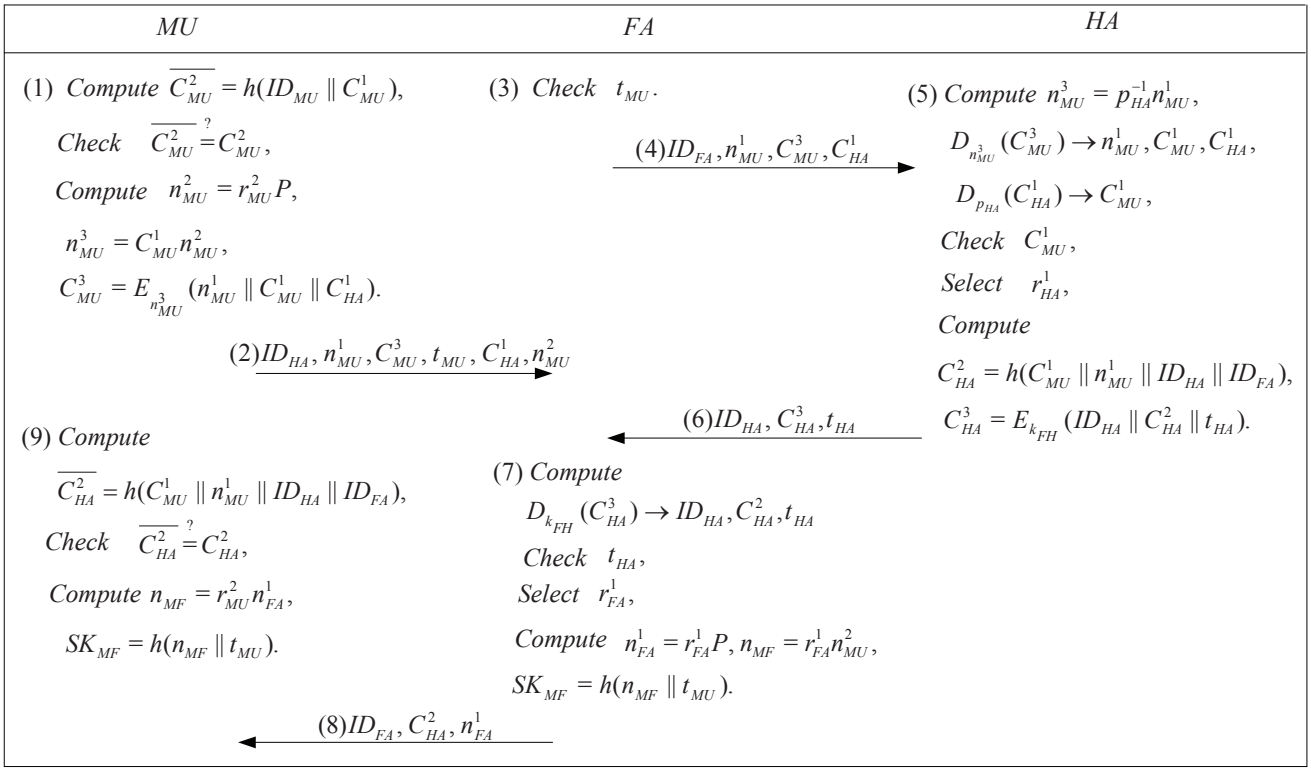


Figure 2: Authentication with key agreement phase of the proposed scheme

a session key with FA via the authentication phase in advance.

1) MU selects a new random number r^i_{MU} and calculates $n^i_{MU} = r^i_{MU} P$, then sends n^i_{MU} to FA.
2) FA first checks whether t^i_{MU} is valid. If it is valid, FA selects a new random number r^i_{FA} and calculates $n^i_{FA} = r^i_{FA} P$, $n^i_{MF} = r^i_{FA} n^i_{MU}$, $SK^{i+1}_{MF} = h(n^i_{MF} || t^i_{MU})$, $C^i_{FA} = h(n^i_{FA} || SK^{i+1}_{MF})$. After computing, FA submits $\{n^i_{FA}, C^i_{FA}\}$ to MU.
3) After receiving the message, MU calculates $n_{MF} = r^i_{MU} n^i_{FA}$, $SK^{i+1}_{MF} = h(n_{MF} || t^i_{MU})$, $\overline{C^i_{FA}} = h(n^i_{FA} || SK^{i+1}_{MF})$ and checks $\overline{C^i_{FA}} \stackrel{?}{=} C^i_{FA}$ to verify FA. If they are equal, FA is authenticated by MU.

Password Change Phase:

When MU's password is expired or leaked, MU may wish to change PW_{MU} for the sake of security. The process of password change phase can be finished without communicating with HA. Detailed steps are described as follows:

1) MU inserts his smart card into a card reader and inputs his identity ID_{MU} and password PW_{MU}. Then, the smart card checks $\overline{C^2_{MU}} = h(ID_{MU} || h(PW_{MU} || r^1_{MU})) \stackrel{?}{=} C^2_{MU}$. If they are equal, MU generates a new random number r^{new}_{MU}.
2) MU computes $C^{2_{new}}_{MU} = h(ID_{MU} \| h(PW^{new}_{MU} \| r^1_{MU}))$, $C^{1_{new}}_{HA} = E_{SK} h(PW^{new}_{MU} \| r^1_{MU})$, $n^{new}_{MU} = h(PW^{new}_{MU} \| r^1_{MU}) r^{new}_{MU} P$.
3) The smart card replaces $\{n^1_{MU}, C^2_{MU}, C^1_{HA}\}$ with $\{n^{new}_{MU}, C^{2_{new}}_{MU}, C^{1_{new}}_{HA}\}$ to finish the password change phase.

6 Security Analysis

In this section, security analysis is carried to confirm that our scheme provides many security properties. The following attacks are based on the assumptions that a malicious attacker $\mathcal{A}$ has completely monitor over the communication channel in authentication and establish the session key phase. So $\mathcal{A}$ can eavesdrop, modify, insert, or delete any message transmitted via public channel [1].

Modification Attack with Smart Card:

1) We assume $\mathcal{A}$ extracts [9, 16] the secret parameters stored in the smart card. No matter $\mathcal{A}$ intends to impersonate FA or MU, he cannot cheat HA successfully. Suppose $\mathcal{A}$ intends to impersonate FA, he can only modify C^3_{MU} by constructing a fraud $\overline{C^3_{MU}}$ since he has no knowledge of r^2_{MU} based on the security of CDL problem. Then, he sends a fraud

$\{ID_{FA},\ \overline{C^3_{MU}},\ C^1_{HA}\}$ to HA. After receiving the message, HA tries to decrypt the $\overline{C^3_{MU}}$ by using its computed decryption key n^3_{MU} but failed. Thus, HA can immediately identify the attack from FA. Similarly, since C^3_{MU} is completely derived from MU, if $\mathcal{A}$ wants to masquerade as MU, he cannot cheat HA successfully, either.

2) To impersonate HA to FA, $\mathcal{A}$ has to generate a legal message $\{ID_{HA},\ C^2_{HA},\ C^3_{HA},\ t_{HA}\}$, where $C^2_{HA} = h(C^1_{MU}||n^1_{MU}||ID_{HA}||ID_{FA})$, $C^3_{HA} = E_{k_{FH}}(ID_{HA}||C^2_{HA}||t_{HA})$, but $\mathcal{A}$ cannot generate correct C^3_{HA} without the knowledge of k_{FH}. Therefore, $\mathcal{A}$ cannot impersonate HA to FA.

3) To impersonate FA to MU, $\mathcal{A}$ must obtain k_{HA} to compute C^3_{HA}, but $\mathcal{A}$ cannot know it. It means any attacker cannot compute the correct C^3_{HA}, $\mathcal{A}$ cannot cheat MU successfully. Since C^3_{HA} directly comes from HA, $\mathcal{A}$ cannot successfully cheat MU by masquerading as HA, either.

Anonymity:

In our scheme, ID_{MU} is protected by two times hash operation consisting of a random number r^1_{MU} chosen by the MU and the two hashed ID_{MU} is well protected by a symmetric encryption algorithm. So, even if $\mathcal{A}$ steals smart card and collects the previous messages transmitted between MU and FA, he cannot derive the real ID_{MU} without the knowledge of a random number r^1_{MU} and MU's password. In other words, our scheme can provide user anonymity.

Mutual Authentication:

1) HA authenticates FA by decrypting $C^3_{MU} = E_{n^3_{MU}}(n^1_{MU}||C^1_{MU}||C^1_{HA})$ with its own secret key p_{HA} to obtain C^1_{MU} and compare with the existing C^1_{MU}, where $n^3_{MU} = p^{-1}_{HA} n^1_{MU}$. In addition, the authentication of HA to MU is completely dependent on the authentication of HA to FA.

2) MU authenticates FA by verifying the computed $\overline{C^2_{HA}} = h(h(PW_{MU}\ ||\ r^1_{MU})\ ||\ n^1_{MU}\ ||\ ID_{HA}\ ||\ ID_{FA})$ with the received C^2_{HA}. At the same time, the authentication of MU to HA is completely dependent on the authentication of MU to FA.

3) FA authenticates HA by decrypting C^3_{HA} by using k_{FH} to obtain $\{ID_{HA},\ C^2_{HA},\ t_{HA}\}$ and check the correctness of the received ID_{HA} and t_{HA}.

Two-Factor Security:

Assume $\mathcal{A}$ steals the MU's smart card and extracts [8, 18] the information $\{ID_{HA},\ n^1_{MU},\ r^1_{MU},\ C^1_{HA},\ C^2_{MU}\}$ stored in it, where

$$\begin{aligned} n^1_{MU} &= r^2_{MU} C^1_{MU} p_{HA} P, \\ C^2_{MU} &= h(ID_{MU}||h(PW_{MU}||r^1_{MU})), \\ C^1_{HA} &= E_{p_{HA}} C^1_{MU}. \end{aligned}$$

At the same time, he may also obtain a message $\{ID_{HA},\ C^3_{MU},\ t_{MU},\ C^1_{HA},\ n^2_{MU}\}$ sent from MU, where

$$\begin{aligned} C^3_{MU} &= E_{n^3_{MU}}(n^1_{MU}||C^1_{MU}||C^1_{HA}), \\ n^2_{MU} &= r^2_{MU} P. \end{aligned}$$

Even though $\mathcal{A}$ can guess a password PW'_{MU}, he cannot compute the correct C^3_{MU}, since it is protected by the random number r^2_{MU} only known by MU, that is, he cannot pass the authentication of HA. Thus, our scheme achieves two-factor security.

Replay Attack:

In the authentication with key agreement phase, though $\mathcal{A}$ intercepts all the messages which are transmitted among MU, FA and HA and replays it to FA, HA, MU respectively. However, $\mathcal{A}$ still cannot compute the correct session key $SK_{MF} = h(n_{MF}||t_{MU})$ due to the usage of two one-time random numbers r^1_{FA} and r^2_{MU} are only held by MU and FA respectively, and timestamp t_{MU} generated by MU. If $\mathcal{A}$ replays the transmitted messages, the receiver can detect the invalid timestamp t_{MU} and terminate the request.

Perfect Forward Secrecy:

Assume $\mathcal{A}$ obtains all of participants' secret keys and previous session keys, he still cannot compromise session key $SK_{MF} = h(n^i_{MF}||t^i_{MU})$. Since $\mathcal{A}$ cannot compute correct n^i_{MF} which is based on the security of CDL and CDH problem, where $r^i_{MU},\ r^i_{FA}$ are different for each session and thus they are not related to previous values. Therefore, our proposed scheme can provide the perfect forward secrecy.

Insider Attack:

In the registration phase, MU only provides his identity ID_{MU} to HA, PW_{MU} is not directly exposed to HA. Any insiders cannot obtain the user's password which is based on the security of one-way hash function. Therefore, our proposed scheme can withstand the insider attack.

Man-in-Middle Attack:

Our proposed scheme can provide mutual authentication among MU, FA and HA. It means that our proposed scheme can withstand the man-in-the middle attack.

Local Password Verification:

In the authentication with key agreement phase, the smart card checks the validity of MU's identity ID_{MU} and password PW_{MU} before logging into FA.

Since $\mathcal{A}$ cannot compute the correct C_{MU}^2 without the knowledge of ID_{MU},and PW_{MU} to pass the verification equation $\overline{C_{MU}^2} = C_{MU}^2$, thus our scheme can avoid the unauthorized accessing by the local password verification.

No Verification Table:
It is obvious that our scheme has no need to store the password or a verification table.

7 Performance and Security Properties Comparison

In this section, we present the comparisons of our scheme with some related schemes in terms of computational cost and functionality. Figure 3 shows the computational cost of our proposed scheme and some other related schemes. Here we mainly focus on the computational cost of the registration and authentication phases because these phases are the principal part of an authentication scheme. Let PM, SE, ASE and H be the number for performing elliptic curve point multiplication, symmetric key encryption or decryption, asymmetric key encryption or decryption and a hash function. Since xor operations require very little computations, we omitted it. From Figure 3 we can see that our proposed scheme is efficient compared with other related schemes. Table 2 lists the functionality comparisons among our proposed scheme and other related schemes. It is obviously that our scheme has many excellent features and is more secure than other related schemes.

8 Conclusion

In this paper, we have shown that Kuo et al.'s scheme is not strong enough against some security weaknesses, such as a vulnerability to insider and verifier attacks while it does not provide local verification. And their password change phase has a loophole. In order to withstand security flaws in Kuo et al.'s scheme, we propose a novel anonymous authentication scheme for wireless communications. Meanwhile, our scheme can resist replay, man-in-middle and can provide anonymity, mutual authentication, perfect forward secrecy and two-factor security. In addition, our scheme is immune to a modification attack with smart cards which has not been considered in other related works. Finally, in comparison with the previously proposed schemes on performance and security prove that our scheme is efficient and is secure against various attacks.

Acknowledgments

The authors would like to thank Prof. Min-Shiang Hwang and the anonymous reviewers for their helpful advice. This paper is supported by the National Natural Science Foundation of China (Grant Nos. 61262057, 61472433).

References

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

[2] 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, 1008-1032, 2013.

[3] 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.

[4] J. B. Hu, H. Xiong, and Z. Chen, "Further improvement of an authentication scheme with user anonymity for wireless communications," *International Journal of Network Security*, vol. 14, no. 5, pp. 297-300, 2012.

[5] H. F. Huang, H. W. Chang, and P. K. Yu, "Enhancement of timestamp-based user authentication scheme with smart card. *International Journal of Network Security*, vol. 16, no. 4, 385-389, 2014.

[6] H. J. Jo, J. H. Paik, and D. H. Lee, "Efficient privacy-preserving authentication in wireless mobile networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1469-1481, 2014.

[7] R. Joos, and A. R. Tripathi,"Mutual authentication in wireless networks," *Technical Report*, 1997.

[8] J. S. Kim, and J. Kwak, "Improved secure anonymous authentication scheme for roaming service in global mobility networks," *International Journal of Security and Its Applications*, vol. 6, no. 3, pp. 45-54, 2012.

[9] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of Advances in Cryptology (CRYPTO'99)*, LNCS 1666, pp. 388-397, 1999.

[10] W. C. Kuo, H. J. Wei, and J. C. Cheng, "An efficient and secure anonymous mobility network authentication scheme," *Journal of Information Security and Applications*, vol. 19, no. 1, pp.18-24, 2014.

[11] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770-772, 1981.

[12] C. C. Lee, S. T. Chiu, and C. T. Li, "Improving security of a communication-efficient three-party password authentication key exchange protocol," *International Journal of Network Security*, vol. 17, no. 1, pp. 1-6, 2015.

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

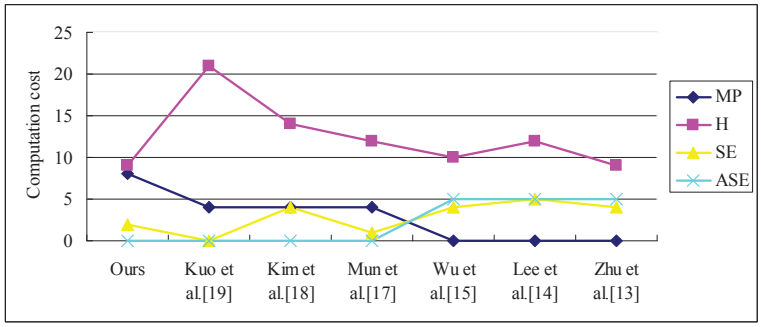


Figure 3: Computational cost comparison

Table 2: Functionality comparison

	Ours	**Kuo et al. [10]**	**Kim et al. [8]**	**Mun et al. [18]**	**Wu et al. [22]**	**Lee et al. [13]**	**Zhu et al. [23]**
Achieve anonymity	Yes	Yes	Yes	No	No	No	No
Provide mutual authentication	Yes	Yes	No	No	No	Yes	No
Provide perfect forward secrecy	Yes	Yes	Yes	Yes	No	No	No
Provide two-factor Security	Yes	Yes	Yes	No	No	No	Yes
Provide local password verification	Yes	No	Yes	No	No	No	No
Resist insider attack	Yes	No	No	No	No	No	No
Resist modification attack with smart card	Yes	No	-	-	-	-	-
Resist replay attack	Yes	Yes	Yes	No	No	Yes	No
Resist man-in-middle attack	Yes	Yes	No	No	Yes	Yes	Yes

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

[15] C. T. Li, and C. C. Lee, "A novel user authentication and privacy preserving scheme with smart cards for wireless communications," *Mathematical and Computer Modelling*, vol. 55, no. 1, pp. 35-44, 2012.

[16] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541-552, 2002.

[17] V. S. Miller, "Use of elliptic curves in cryptography," *Proceeding on Advances in Cryptology-CRYPTO'85*, Springer-Verlag, New York, pp. 417-426, 1985.

[18] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun, and H. H. Choi, "Enhanced secure anonymous authentication scheme for roaming service in global mobility networks," *Mathematical and Computer Modelling*, vol. 55, no. 1-2, pp. 214-222, 2012.

[19] A. Prakash, and R. Mukesh, "A biometric approach for continuous user authentication by fusing hard and soft traits," *International Journal of Network Security*, vol. 16, no. 1, pp. 65-70, 2014.

[20] N. Sklavos, and X. M. Zhang, "Wireless security and cryptography: specifications and implementations," *CRC-Press*, ISBN: 084938771X, 2007.

[21] Y. Wang, D. S. Wong, and L. S. Huang, " One-pass key establishment protocol for wireless roaming with user anonymity," *International Journal of Network Security*, vol. 16, no. 2, pp. 129-142, 2014.

[22] C. C. Wu, W. B. Lee, and W. J. Tsaur, "A secure authentication scheme with anonymity for wireless communications," *IEEE Communications Letters*, vol. 12, no. 10, pp. 722-723, 2008.

[23] J. M. Zhu, and J. F. Ma, "A new authentication scheme with anonymity for wireless environments," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 230-234, 2004.

Yan-Rong Lu received the M.S. degree in cryptography from Xidian University of China, Xi'an, China, in 2012. She is currently a Ph.D. student in Beijing University of Posts and Telecommunications, Beijing, China. Her research interests is focused on information security and cryptography, in particular, cryptographic protocols.

Xiao-Bo Wu received her B.S. degree in computer science from Yantai Normal College (China) in June 2002, M.S. degree in computer science and technology from Dalian University of Technology (China) in June 2008. Her current research interest includes protocols analysis, information security and cryptography.

Xiao-Dong Yang received B.S. degree in applied mathematics from Northwest Normal University (China) in June 2002, M.S. degree in applied mathematics from Tongji University (China) in June 2005 and the Ph.D degree in information security from Northwest Normal University (China) in June 2010. He is also an associate professor in information and computer science at Northwest Normal University. His current research interest includes proxy re-signature and its application, wireless sensor network security, information security and cryptography.

Self-verifiable Secret Sharing Scheme with Locatability for Halftone Images

Yi-Hui Chen[1], Panyaporn Prangjarote[2], and Chih-Yang Lin[2]
(Corresponding author: Yi-Hui Chen)

Department of Applied Informatics and Multimedia, Asia University[1]
Department of Computer Science & Information Engineering, Asia University [2]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
(Email: chenyh@asia.edu.tw)

(Received Otc. 2, 2013; revised and accepted Dec. 6, 2013)

Abstract

Visual cryptography is an image secret sharing scheme used to encrypt the secret image into several meaningless share images. Later on, the secret image can be obtained while stacking the share images with no computations. However, the secret image cannot be completely reconstructed. In particular domains, it is intolerant any distortions, such as medical image, legitimate, artist, etc. With low computations, lossless image secret sharing is proposed to help completely reconstruct the secret image. During the transmission, it is a risk that some bits in a share image are lost. As a result, the secret image cannot to be completely reconstructed. To meet requirement, it desires a way to authenticate whether the bits are lost. In this paper, a self-authentication mechanism for image secret sharing scheme is proposed. The experiments provide the positive data to show the feasibility of the proposed scheme.

Keywords: Authentication mechanism, image secret sharing, lossless reconstruction

1 Introduction

Image secret image sharing scheme is widely used in secret image protection. Two major types of image secret sharing are visual cryptographic scheme and polynomial-based secret image sharing scheme. Visual cryptography [3, 8, 9, 10, 13] encrypts the secret image into noise-like share images. Later on, the secret image can be obtained while superimposing the share images through a human visual system with no assistance of computations. However, the stacked secret image is lossy and insufficient in some certain domains, such as arts, legislation, medical images, etc. because any distortions are intolerant. Polynomial-based secret sharing scheme is the lossless secret sharing schemes [1, 2, 11, 12, 14, 17, 20], traced back to the general (t, n)-threshold secret sharing proposed by Shamir in 1997 [14, 15, 16], are introduced in secret image protection with reversibility. That is, the secret image can be decrypted by using Lagrange interpolation only if more than t shares are collected; otherwise, the original secret image can never be obtained. However, it requires handle large computations. With lower computation loads, a new application in sharing secret image [4, 18, 19] has proposed to losslessly reconstruct the secrets with Boolean operations.

After reconstructing the secrets, a mechanism used to verify the fidelity of the reconstructed secret, so-called authentication mechanism. As for (t, n)-threshold image secret sharing schemes, a significant amount of researches [1, 2, 11, 12, 17, 20] provided authentication abilities to check whether the reconstructed secret image is authentic. In 2004, Lin and Tsai [11] provided an authentication mechanism on (t, n)-threshold image secret sharing scheme using parity check. Next, Yang et al. [20] improved the performances of Lin and Tsai's scheme to provide more precise authentication results, which can detect 50% of fake blocks, and lossless quality of the reconstructed image. Chang et al. [1] provides 93% detection rate of fake blocks by using Chinese Remainder Theorem (CRT). In 2010, Chang et al. [2] provided a remedy version to recovery the inauthentic areas to prevent from bits lost during data transmission over the Internet.

As for the visual cryptography, Naor and Pinkas [13] are the pioneers of visual authentication using visual cryptography. Horng et al. [8] point out the possible cheatings in visual cryptographic schemes and claim that authentication mechanism can prevent from the cheating. Later on, Hu and Tzeng [10] proposed a cheating prevention method in visual cryptography. In addition, another authentication mechanism is proposed by schemes [5, 6] to extract the authentication image by stacking the shares at different positions [6, 7, 8, 9, 10, 11]. For low computation loads, but lossless reconstruction ability, Chen and Lin [7] proposed an authentication for lossless image secret sharing scheme.

Generally, the authentication mechanism considers the authentication codes as one of the factors for generating the image shares and then extracts the authentication codes to authenticate whether it not a fake one. That is, the reconstructed image will be judged as inauthentic if the extracted authentication codes are not equal to the pre-defined authentication image or the hashing results. The authentication codes are treated as an authentication image, like a pre-defined watermark or a logo, or hashing results. The pre-defined authentication image might raise the potential risk that attackers generate the fake shares according to the authentication image while collecting some valid shares. The hash results are generated by hashing the secret image to be the authentication codes with a pre-shared key. It works but a bit unexpectedly lost of the share during transmission suffers from the hash codes being quite different, which do not have ability to indicate where is inauthentic. In this paper, an efficient authentication mechanism is proposed to locate where is inauthentic. The rest of this paper is organized as follows. In Section 2, we propose a self-verifiable scheme to locate the inauthentic areas. Experimental results are shown in Section 3. The conclusions are made in Section 4.

2 The Proposed Scheme

We propose an efficient visual secret sharing scheme complied with authentication ability, including three procedures: encryption and embedding, decryption and verification. In the encryption phase, a secret image is converted into three seemingly random pictures using exclusive-or operation. In the decryption phase, all three shares are stacked together to reveal the secret image. The verification procedure is helpful to verify whether the reconstructed secret image is not modified or some bits lost during the transmissions.

2.1 Encrypting the Secret Image

A given secret image S with the sized of $w \times h$ pixels, is first divided vertically into two areas, denoted as L and R. As shown in Figure 1, we define two pixels as a pixel pair in S, namely $s_{(i,j)}$ and $s_{(i,w-j+1)}$, which are located at (i, j) and $(i,w-j+1)$, respectively. Pixels $s_{(i,j)}$ and $s_{(i,w-j+1)}$ are related because they are encoded and decrypted at the same time.

To generate authentication signals, the proposed scheme exploits the secret information identical to the original secret images. For simplicity, the encrypted pixels in the share images S_1, S_2, and S_3 should be denoted as $s_{1(i,j)}$, $s_{2(i,j)}$, $s_{3(i,j)}$, $s_{1(i,w-j+1)}$, $s_{2(i,w-j+1)}$ and $s_{3(i,w-j+1)}$, as illustrated in Figure 2.

The encryption and embedding phase consists of two main steps.

Step1. Generate two random matrices with two secret key sk_1 and sk_2, which are size of $w/2 \times h$ pixels (a half of secret images) consisting of binary integer: 0 or 1. Each element in the first matrix is represented by R_1 while the other random matrix is R_2. The pixels in R_1 and R_2 located at (i, j) are denoted as $R_{1(i,j)}$ and $R_{2(i,j)}$, respectively. During the authentication procedure, the keys R_1 and R_2 must be kept to help in indicating the inauthentic regions.

Step2. We design new encryption operations as follows, to encrypt every secret pixel into random pixels.

$$s_{1(i,j)} = R_{1(i,j)} \oplus s_{(i,j)} \quad (1)$$

$$s_{2(i,j)} = R_{2(i,j)} \quad (2)$$

$$s_{3(i,j)} = R_{1(i,j)} \oplus R_{2(i,j)} \quad (3)$$

$$s_{1(i,w-j+1)} = R_{1(i,j)} \oplus s_{(i,w-j+1)} \quad (4)$$

$$s_{2(i,w-j+1)} = R_{1(i,j)} \oplus R_{2(i,j)} \oplus s_{(i,j)} \quad (5)$$

$$s_{3(i,w-j+1)} = R_{2(i,j)} \oplus s_{(i,w-j+1)} \quad (6)$$

After encryption process, three shares are generated, in which authentication code can be embedded simultaneously. That is, the three image shares contain a duplicated copy of the secret image (i.e., authentication signals), which can be used to verify the integrity of the decrypted secret image.

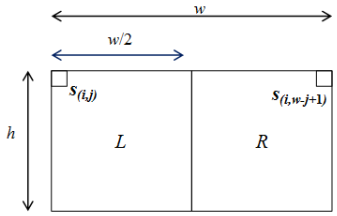
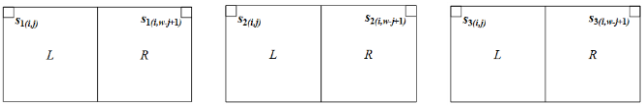


Figure 1: Two corresponding secret pixels



(a) Share S_1 (b) Share S_2 (c) Share S_3

Figure 2: Six related pixels in three shadows

2.2 Decrypting the Secret Images

Three shares are collected at the receiver side to decrypt the secret image. Note that six pixels, i.e., $s_{1(i,j)}$, $s_{2(i,j)}$, $s_{3(i,j)}$, $s_{1(i,w-j+1)}$, $s_{2(i,w-j+1)}$ and $s_{3(i,w-j+1)}$, are treated as a group used to decrypt two secret images. According to Equations (7) and (8), the pixel located at (i, j) in secret image can be decrypted. Also, the pixel located at $(i,w-j+1)$ can be decrypted with Equations (9) and (10).

$$s'_{(i,j)} = s_{1(i,j)} \oplus s_{2(i,j)} \oplus s_{3(i,j)} \quad (7)$$

$$s'_{(i,j)} = s_{1(i,w-j+1)} \oplus s_{2(i,w-j+1)} \oplus s_{3(i,w-j+1)}, \quad (8)$$

$$s'_{(i,w-j+1)} = s_{1(i,j)} \oplus s_{2(i,w-j+1)} \oplus s_{3(i,w-j+1)}, \quad (9)$$

$$s'_{(i,w-j+1)} = s_{2(i,j)} \oplus s_{3(i,j)} \oplus s_{1(i,w-j+1)}, \quad (10)$$

2.3 Verifying the Decrypted Secret Image

The authentication mechanism is aimed at checking whether the decrypted secret image is an authentic one. The ability of authentication has a limitation: the authentication mechanism can detect whether it is modified only when just one of the six pixels is lost or modified. The authentication mechanism is used to judge whether the pixels located at (i, j) and (i,w-j+1) in decrypted image are authentic. If the pixels $s'_{(i,j)}$ and $s'_{(i,w-j+1)}$ are decrypted with Equations (7) and (9) are equal to that are with Equations (8) and (10), the decrypted pixels are authentic; otherwise, inauthentic.

3 Experimental Results

Three different halftone images: "Lena", "F16", and "Baboon" are used to demonstrate the effectiveness of authentication ability and to show the efficiency of the proposed VSS scheme. The size of test images in the experiments is 512×512 pixels, and the halftone images are given in Figure 3.

3.1 Experimental Results of Sharing and Embedding Using Binary Images

Since we adopt (3, 3) Boolean-based VSS for encryption and decryption, our method can successfully encrypts the secret image into an intelligible version, with the help of only two random matrices. Although we embed an authentication image which the size is as large as secret image, Figure 4 shows that the size of shares is still preserved. That is, pixel expansion problem is removed. Furthermore, the reconstructed images are identical to the original secret images after decryption; therefore, the proposed scheme achieves lossless secret reconstruction. The lossless reconstructed images and their related authentication images are shown in Figure 5.

3.2 Experimental Results of Binary Image Authentication

We test our proposed image authentication algorithms by modifying shares images using bit flipping operation. As an example, Figure 6(a) shows the first image share of "Lena" and this share was modified by 30×30 fake pixels drawn as dot rectangle in Figure 6(b). Authentication result, shown in Figure 6(c), reveals authentic pixels in the white region while black region indicates unauthentic pixels. We can conclude that the verifying process of the proposed scheme is highly effective because it is able to fully detect tampered region.

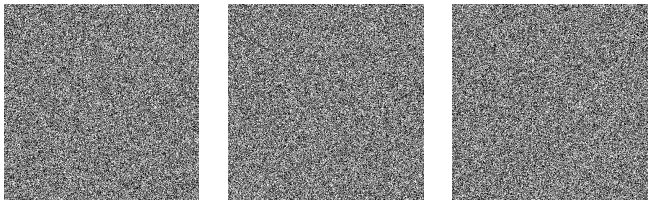
As can be seen the wrong tampered region in circle dotted line in Figure 6(c), this result caused by a crossing problem, which is that some non-modified pixels were detected as modified pixels. Therefore, the number of un-tampered pixels will be as twice as that of tampered pixels.

4 Conclusions

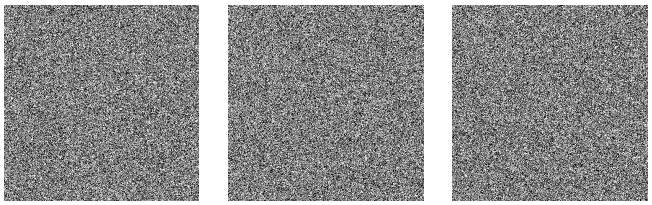
In this paper, a new idea in secret sharing based on visual secret sharing with a helpful technique called authentication is proposed. The proposed scheme applies Boolean-based VSS to generated desirable noise images from a given secret image. Then, after combining image shares, the image quality of reconstructed image is still the same as the original secret image. Moreover, the authentication is applied in the secret sharing scheme to authenticate which are the authentic bits. However, the authenticate result are suffered from a crossing problem. In the future, we are planning to resolve the problem.

(a) Lena (b) F16 (c) Baboon

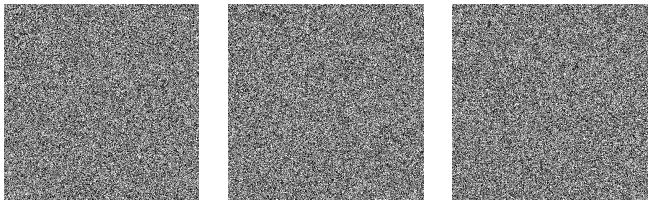
Figure 3: Secret images



(a) share1 (b) share2 (c) share3

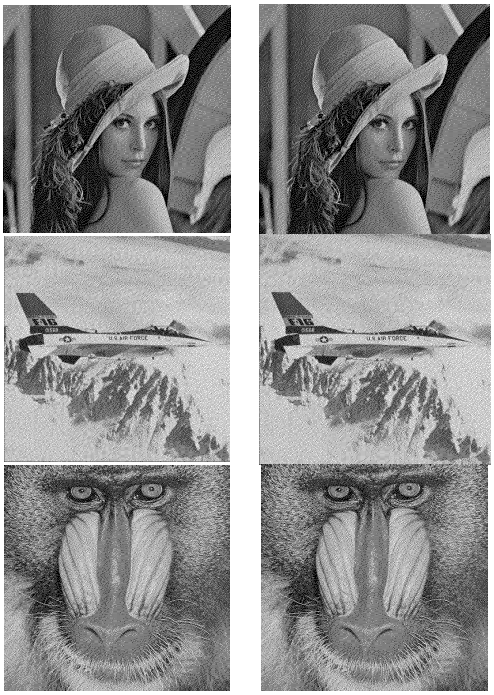


(d) share1 (e) share2 (f) share3



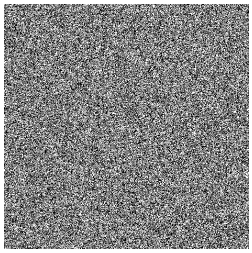
(g) share1 (h) share2 (i) share3

Figure 4: Encryption results. (a)-(c) the shared images of Lena, (d)-(f) F16 and (g)-(i) Baboon image.

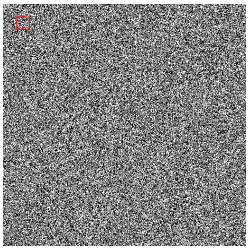


(a) Reconstructed *S'* (b) Reconstructed *A'*

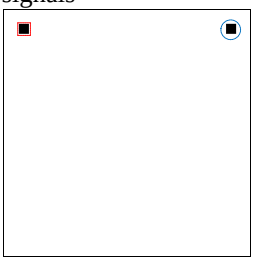
Figure 5: Reconstructed images



(a) Share1 with authentication signals



(b) The modified share1



(c) Result with tampered pixels detected and marked in black

Figure 6: Authentication result

Acknowledgments

Many thanks for the support by National Science Council of Taiwan with the No. 102-2221-E-468-009.

References

[1] C. C. Chang, Y. P. Hsieh, and C. H. Lin, "Sharing secrets in stego images with authentication," P*attern Recognition*, vol. 41, no. 10, pp. 3130-3137, 2008.

[2] C. C. Chang, Y. H. Chen, and H. C. Wang, "Meaningful secret sharing technique with authentication and remedy abilities," *Information Sciences*, vol. 181, no. 14, pp. 3073-3084, 2011.

[3] T. H. Chen and K. H. Tsao, "User-friendly random-grid-based visual secret sharing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 11, pp. 1693-1703, 2011.

[4] T. H. Chen and C. S. Wu, "Efficient multi-secret image sharing based on boolean operations," *Signal Processing*, vol. 91, no. 1, pp. 90-97, 2011.

[5] Y. C. Chen, D. S. Tsai, and G. Horng, "a new authentication based cheating prevention scheme in naor–shamir's visual cryptography," *Journal of Visual Communication and Image Representation*, vol. 23, no. 8, pp. 1225-1233, 2012.

[6] Y. H. Chen, "An efficient authentication mechanism for (2, 2)-visual cryptography scheme," *Journal of Computers*, vol. 22, no. 4, pp. 3-10, 2012.

[7] Y. H. Chen and P. Y. Lin, "An authentication mechanism for secret sharing using a boolean operation," *Journal of Electronic Science and Technology*, vol. 10, no. 3, pp. 195-198, 2012.

[8] G. Horng, T. Chen, and D. s. Tsai, "Cheating in visual cryptography," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 219-236, 2006.

[9] Y. C. Hou and Z. Y. Quan, "Progressive visual cryptography with unexpanded shares," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 11, pp. 1760-1764, 2011.

[10] C. M. Hu and W. G. Tzeng, "Cheating prevention in visual cryptography," I*EEE Transactions on Image Processing*, vol. 16, no. 1, pp. 36-45, 2007.

[11] C. C. Lin and W. H. Tsai, "Secret image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 73, no. 3, pp. 405-414, 2004.

[12] P. Y. Lin and C. S. Chan, "Invertible secret image sharing with steganography," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1887-1893, 2010.

[13] M. Naor and A. Shamir, "Visual cryptography," in *Advances in Cryptology-Eurocrypt' 94*, LNCS 950, pp. 1-12, Springer-Verlag,1995.

[14] S. Rao Y V and C. Bhagvati, "CRT based threshold multi secret sharing scheme," *International Journal of Network Security*, vol. 16, no. 3, pp. 194-200, 2014.

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

[16] Y. L. Tian, C. G. Peng, and J. F. Ma, "Publicly verifiable secret sharing schemes using bilinear pairings," *International Journal of Network Security*, vol. 14, no. 3, pp. 142-148, 2012.

[17] G. Ulutas, M. Ulutas, and V. V. Nabiyev, "Secret image sharing scheme with adaptive authentication

strength," Pattern Recognition Letters, vol. 34, no. 3, pp. 283-291, 2013.
[18] D. Wang, L. Zhang, N. Ma, and X. Li, "Two secret sharing schemes based on boolean operations," *Pattern Recognition*, vol. 40, no. 10, pp. 2776-2785, 2007.
[19] X. Wu and W. Sun, "Random grid-based visual secret sharing with abilities of OR and XOR decryptions," *Journal of Visual Communication and Image Representation*, vol. 24, no. 1, pp. 48-62, 2013.
[20] C. N. Yang, T. S. Chen, K. H. Yu, and C. C. Wang, "Improvements of image sharing with steganography and authentication," J*ournal of Systems and Software*, vol. 80, no. 7, pp. 1070-1076, 2007.

Yi-Hui Chen received B.S. and M.S. degrees in information management from the Chaoyang University of Technology in 2001 and 2004, respectively. In 2009, she earned her Ph.D. degree in computer science and information engineering at the National Chung Cheng University. From 2009 to 2010, she worked with Academia Sinica as a post-doctoral fellow. Later, she worked at IBM's Taiwan Collaboratory Research Center as a Research Scientist. She is now an assistant professor with the Department of Applied Informatics and Multimedia, Asia University. Her research interests include image processing, watermarking, steganography, and XML techniques.

Panyaporn Prangjarote received her B.S. degree in computer science from Naresuan University, Phitsanuloke, Thailand in 2000 and M.S. in computer science from Asian Institute of Technology (AIT), Bangkok, Thailand in 2003. She is currently a Ph.D. student in computer science and information engineering at Asia University, Taichung, Taiwan.

Chih-Yang Lin received the B.S. degree in computer science and information engineering from Tung-Hai University, Taichung, in 1998, the master degree in information management from National Chi-Nan University, Nantou, in 2000. In 2006, he received a Ph.D. degree from Dept. Computer Science and Information Engineering at National Chung-Cheng University, Chiayi. After graduated, he servered in Advanced Technology Center of Industrial Technology Research Institute of Taiwan (ITRI) from 2007 to 2009. Then, he joined the Institute of Information Science (IIS), Academia Sinica, as a postdoctoral fellow. Currently, he is an Assistant Professor in the Department of Computer Science and Information Engineering, Asia University. His research interests include computer vision, digital rights management, image processing, and data mining.

A New Method for Computing DLP Based on Extending Smooth Numbers to Finite Field for Ephemeral Key Recovery

R. Padmavathy[1] and Chakravarthy Bhagvati[2]
(Corresponding author: R. Padmavathy)
Department of Computer Science and Engineering, National Institute of Technology Warangal, India[1]
Department of Computer and Information Sciences University of Hyderabad, Hyderabad, Andhra Pradesh, India[2]
(Email: r_padma3@rediffmail.com, chakcs@uohyd.ernet.in)

(Received Aug. 3, 2012; revised and accepted Aug. 3, 2013)

Abstract

In this paper, new algorithms to solve certain special instances of the Discrete Logarithm Problem (DLP) is presented. These instances are generally considered hard in literature. If a cryptosystem is based on a prime p such that $p-1$ is either $2q$ with q a prime; or 2ρ where $\rho = \gamma_1\gamma_2 \ldots \gamma_k q$ with γs being small prime factors and q a large prime factor, and the exponent is chosen in the middle of the group (or a prime-order subgroup), we show that it is vulnerable. In other words, the attacks proposed in this paper are analogous to the attacks for factoring large numbers when the factors lie near the square-root. The main idea is to generalize the concept of a smooth number and extend it over factor bases and multiplicative groups Z_p^*. We show that for careful selection of factor bases, patterns form in the distribution of such generalized smooth numbers which may be exploited in the attacks. Our algorithms are empirically tested on several hundred problems with sizes ranging from 100 – 1024 bits and the average running times show the performance of the newly developed attacks. Also, the key recovery attack proposed on Chang and Chang novel three party simple key exchange protocol is mounted by recovering the ephemeral keys. The ephemeral keys are recovered by solving DLP using the new algorithms proposed in the present study.

Keywords: Chang-Chang password key exchange protocol, cryptanalysis, discrete logarithm problem, key recovery attack, smooth numbers

1 Introduction

In this paper, we propose new methods to solve the discrete logarithm problem (DLP) for cases that have been considered hard in literature. The discrete logarithm problem forms the basis for the popular El-Gamal public key cryptosystem [5], Diffie-Hellman key exchange [4] and several digital signature schemes. For a given prime number p, a generator $g \in Z_p^*$ and an element $y \in Z_p^*$, the problem of finding an x $(0 \leq x \leq p-2)$ such that $g^x \equiv y \bmod p$ is known as the discrete logarithm problem. DLP is also defined over other groups such as the multiplicative group of F_{2^m} and the collection of points defined by an elliptic curve over a finite field. In particular, we show that our method is effective on the so-called *safe primes* of the form $p = 2q+1$ and for large exponents $(\approx q)$.

Several methods of solving DLP have been proposed in literature. Shanks' baby-step-giant-step [6] is a well-known deterministic algorithm with both time and space complexities of $O(\sqrt{n})$ where n is the order of the group. The probabilistic Pollard-Rho method [17] also has a $O(\sqrt{n})$ running time but avoids the large space requirements. Another variant of the Pollard-Rho algorithm, called the Pollard-Lambda, solves the DLP in a time of $O(\sqrt{w})$ if the exponent is known to lie in an interval of width w [17]. One of the more popular algorithms is the Pohlig-Hellman method [16] which reduces the DLP to searching over small subgroups and using the Chinese Remainder Theorem to combine the results to solve the DLP. Pohlig-Hellman is a very effective method when the order of the group Z_p^*, $p-1$, has no large factors.

Index Calculus methods are some of the best methods for solving DLP if there is more structure in the group than merely the set of elements and a binary operation [3, 9, 20, 21]. Specifically, if the group elements (*smooth numbers*) can be expressed as a product of a smaller subset of group elements (called the *factor base*), then index calculus methods allow for solving the DLP in sub-exponential time. Recently some improvements over ICM are narrated in [12] and [14]. The DLP can be solved

more efficiently through trap-doors when additional information is available. When the element y is restricted to a small subgroup, DLP can be solved in reduced time using any of the known exponential time algorithms. Van Oorschot and Wiener [11] presented an attack when the exponent x is small and suggested the use of prime-order subgroups, safe primes and large exponents to avoid their attack. A prime p is considered safe if $p-1$ does not have any small factors or has at least one large factor. Later, Lim and Lee [7] proposed an attack on prime-order subgroups.

Today, it is not sufficient to chose a large p for the security of any cryptosystem using the DLP. DLP over Z_p^* has been solved for 120 digits ($\approx$ 360 bits) in the general case [18], and for nearly 607 bits over F_{p^n} [19]. It has led to a recommendation that a problem size of 1024 bits be used and that p be chosen such that $p-1$ has at least one large prime factor for a cryptosystem to be considered safe.

In this paper, we develop two new algorithms for solving the DLP. The first is for solving the DLP on safe primes, i.e., primes of the form $p = 2q+1$, and the second for large prime-order subgroups. The first algorithm is based on extending the concept of smooth numbers to finite fields, and on their distributions for primes of the form $p = 2q+1$. The second algorithm combines the ideas proposed by Lim and Lee [7] and the first algorithm to solve the DLP over prime-order subgroups. Also, the ephemeral keys in Chang and Chang novel three party simple password key exchange protocol are recovered by solving the DLP using the techniques based on the above two algorithms. Additionally it is shown that the recovered ephemeral keys are used to mount the key recovery attack on the Chang and Chang protocol.

The rest of the paper is organized as follows. Section 2 defines smooth numbers over Z_p^* and describes their distributions for safe primes, Section 3 presents the new techniques to solve DLP. Section 4 shows the experimental results, Section 5 briefs the key recovery attack on Chang and Chang password key exchange protocol while Section 6 presents our conclusions.

2 Smooth Numbers and their Distributions for Certain Primes

An integer is called *B-smooth* if it has no prime factors larger than B. For example, the number 48 is 3-smooth because its prime factors are only 2 and 3. An analogous definition proposed in the present study for smoothness over Z_p^* is as follows.

Definition 1. *Let Z_p^* be a multiplicative group of a prime field p and let FB be a factor base ($FB \subset Z_p^*$). Then an element $s \in Z_p^*$ is said to be FB-smooth over Z_p^* iff*

$$s \equiv \prod_{b_i \in FB} {b_i}^{c_i} \bmod p, c_i \geq 0 \tag{1}$$

The above definition is equivalent to the conventional definition of smooth numbers (B smooth in this case). In general, the factor base in conventional definition consists of all primes less than an upper bound B. But the FB in the proposed definition may now contain *an arbitrary set of primes*. The importance of this proposed definition is analyzed in the rest of the paper.

Let S_p^B denote the set of all FB-smooth numbers over Z_p^*. Let b be an element of Z_p^*. We define a smoothness function as

$$F_s(b) = \begin{cases} 1 & \text{if } b \in S_p^B \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Let $K = \{x : x \in Z_p^* \text{ and } 1 \leq x \leq \frac{p-1}{2}\}$ $L = \{x : x \in Z_p^* \text{ and } \frac{p-1}{2} < x \leq p-1\}$. and

$$X = \{F_s(b), b \in K\} \tag{3}$$

$$Y = \{F_s(b), b \in L\} \tag{4}$$

Note, that the cardinality of the sets X and Y is the same.

Let w_1 represents the string of 0's and 1's, which is obtained by writing the elements of the set X in order and w_2 represents a similar string obtained from the set Y.

Let a_i be the i^{th} element of w_1 and b_j be the j^{th} element of w_2.

Definition 2 (Reverse Pattern). *w_2 is defined as the reverse of w_1 and denoted as $w_2 = w_1^R$ if $a_i = b_{p-i}, \forall a_i$.*

Definition 3 (Complement Pattern). *w_2 is defined as the complement of w_1 and written as $w_2 = w_1^C$ if $a_i = b'_{p-i}, \forall a_i$ where $'$ denotes complement operation.*

Definition 4 (Palindrome Pattern). *Let w denote the string w_1w_2. Then w exhibits a palindrome pattern if w_2 is the reverse of w_1.*

Definition 5 (Inverse Palindrome Pattern). *Let w denote the string w_1w_2. Then w exhibits an inverse palindrome pattern if w_2 is the complement of w_1.*

Theorem 1. *Let p be a prime with $p-1 = 2q$ where q is a prime. Let the factor base FB be chosen such that it contains elements of order $(p-1)/2$, i.e., of order q. In such a case, the distribution of FB-smooth numbers over Z_p^* exhibits an inverse palindrome pattern.*

Proof.

1) Let $x \in Z_p^*$ be an element of order (q) and $Q = -x \bmod p$. Then $x^q \equiv 1$ and $Q^q \equiv -1$ which means that Q is a generator and non-residue. Therefore, if $x \in K$ and $F_s(x) = 1$ then $Q \in L$ and $F_s(Q) = 0$; and if $x \in L$ and $F_s(x) = 1$ then $Q \in K$ and $F_s(Q) = 0$.

2) Let x be an element of order $(2q)$. Then x is a generator.

$$x^{\frac{p-1}{2}} \equiv -1; Q^{\frac{p-1}{2}} \equiv 1.$$

That is, Q is an element of order (q). Therefore, if $x \in K$ and $F_s(x) = 0$ then $Q \in L$ and $F_s(Q) = 1$; and if $x \in L$ and $F_s(x) = 0$ then $Q \in K$ and $F_s(Q) = 1$.

3) Let $x \in K$ be an identity element; then, Q is an element of order 2 and $Q \in L$. Therefore, if $x \in K$ and $F_s(x) = 1$, then $Q \in L$ and $F_s(Q) = 0$.

1), 2) and 3) prove that w_1w_2 exhibits an *inverse palindrome pattern.* □

It may also be noted that if FB, the factor base, contains only elements of order 2, then w_1w_2 represents a *palindrome pattern.*

Theorem 2. *Let p be a prime with $p-1 = 2q_1q_2$ where q_1 and q_2 are primes. Let the factor base FB be chosen such that it contains elements of order (q_1q_2). Then the set S_p^B contains elements of order $(q_1), (q_2)$ and (q_1q_2) and the distribution of FB- smooth numbers over Z_p^* exhibits an inverse palindrome pattern.*

Proof.

1) Let $x \in Z_p^*$ be an element of order (q_1q_2) and $Q = -x \bmod p$. Then,

$$x^{q_1q_2} \equiv 1; Q^{q_1q_2} \equiv -1.$$

That is, Q is a generator.

2) Let $x \in Z_p^*$ be an element of order (q_1) and $Q = -x \bmod p$. Then,

$$x^{q_1} \equiv 1; Q^{q_1} \equiv -1.$$

It shows that Q is an element of order $(2q_1)$. Similarly, if x is an element of order (q_2), then Q is an element of order $(2q_2)$.

From 1) and 2), it may be seen that if $x \in K$ and $F_s(x) = 1$, then $Q \in L$ and $F_s(Q) = 0$; if $x \in L$ and $F_s(x) = 1$, then $Q \in K$ and $F_s(Q) = 0$.

3) Let $x \in Z_p^*$ be an element of order $(2q_1q_2)$ and $Q = -x \bmod p$. Then,

$$x^{q_1q_2} \equiv -1; Q^{q_1q_2} \equiv 1.$$

As $q_1 \neq q_2$ and the order $(2q_1)$ and $(2q_2)$ are even, Q is an element of order (q_1q_2).

4) Let x be an element of order $(2q_1)$ and $Q = -x \bmod p$. Then,

$$x^{q_1} \equiv -1; Q^{q_1} \equiv 1.$$

Thus, Q is an element of order (q_1). Similarly, it may be shown that Q is an element of order (q_2) if x is an element of order $(2q_2)$.

From 2) and 3), it may be seen that if $x \in K$ and $F_s(x) = 0$, then $Q \in L$ and $F_s(Q) = 1$; if $x \in L$ and $F_s(x) = 0$, then $Q \in K$ and $F_s(Q) = 1$.

5) Let $x \in K$ be an identity element and $Q = -x \bmod p$, then $Q \in L$ and is an element of order (2). Therefore, if $x \in K$ and $F_s(x) = 1$, then $Q \in L$ and $F_s(Q) = 0$.

It is proved from the above cases that w_1w_2 exhibits an *inverse palindrome pattern.* □

Corollary 1. *Let FB contain only elements of order $(2q_1)$. Then, S_p^B consists of elements of order $(2), (q_1)$ and $(2q_1)$ and the distribution of $FB-$smooth numbers exhibits a palindrome pattern.*

Corollary 2. *Let FB contain only elements of order $(2q_2)$. Then, S_p^B consists of elements of order $(2), (q_2)$ and $(2q_2)$ and the distribution of $FB-$smooth numbers exhibits a palindrome pattern.*

Both the above corollaries can be easily proven from 2) and 4) of Theorem 2. Similarly, it may also be shown that if FB contains only elements of order (q_1) and (q_2), the distribution of FB- smooth numbers will show no discernible patterns.

2.1 Motivation and Algorithms to Solve the DLP

The motivation to develop new techniques to solve DLP is based on the results obtained in the previous section. The following equations show the relationship between the discrete logarithms of an element, say y and $-y$, in inverse palindrome pattern.

$$x = q \pm 2v \bmod (p-1) \text{ when } y \text{ is of order } 2q \quad (5)$$

$$v = (x \pm q)/2 \bmod q \text{ when } y \text{ is of order } q \quad (6)$$

where q is the order of the subgroup; and, v is the discrete logarithm of $-y$ if x is that of y and vice-versa. The following Algorithm 1 presents the calculation of the DLP using the above equations.

Algorithm 1 Computation of DLP

INPUT: Problem of size p,factors of $p-1$, generator of order $2q$ and y.
OUTPUT: The exponent x.

```
Find the order of y
if y is of order 2q then
    Find the generator g_q of order q from g
    Compute logarithm v of −y using Pollard lambda
    Apply Equation 5 and obtain x
else
    if y is of order q then
        Compute logarithm v of −y using Pollard lambda
        Apply Equation 6 and obtain x
    end if
end if
```

Similarly, the exponents within the approximate distance $\frac{q-1}{2} \pm d$ can be solved in reduced time. To solve the logarithm of y, i.e., $(q-1)/2 \pm d$, the logarithm of $-y$,

i.e., $-1 \pm 2d \mod p-1$ to be solved. Since,

$$\begin{aligned} x &= q \pm 2v \mod p-1 \\ &= q \pm 2((q-1)/2 \pm d) \mod p-1 \\ &= q+q-1 \pm 2d \mod p-1 \\ &= 2q-1 \pm 2d \mod 2q. \end{aligned}$$

The logarithm of $-y$, which could be solvable in minimum computational power is vulnerable. The following section presents the study on the exponents in a specific range to solve the logarithm in reduced cost.

3 Computing the Discrete Logarithm in a Given Range

In the current technology, it is considered infeasible to compute the DLP in a group of order $\approx$ 1024 bits. Van Oorshot and Wiener [11] analyzed the computation of the DLP for an exponent of size $\approx$ 160 bits combined with a random prime p of size $\approx$ 1024 bits. The algorithm is the combination of Pohlig-Hellman and Pollard-Lambda method to solve the DLP with the above constraints and it works as follows. Let $y = g^x$ be an element of a group G of order $n = zQ$, where $z = B_r$ is the product of smooth factors, and has bit length approximately k. Compute V where $V = x \mod z$, by a partial Pohlig-Hellman decomposition. Write $x = Az + V$, where $0 \leq V < z$ with A as yet unknown. Then $y = g^x = g^{Az+V}$. Now $A \in [0, 2^c]$, where $c = u-k$ bits of x remain unknown after finding V. Computing g^V and $y^* = y/g^V = g^{Az} = h^A$, where $h = g^z$ is known. Now V to be computed from the lambda method. Since A and V are known, x can be calculated as $x = Az+V$.They suggested the use of prime order subgroup along with short exponents or the use of safe primes.

Latter, Lim and Lee [7] investigated the DLP in a prime order subgroup of order $\approx$ 160 bits for p of size $\approx$ 1024 bits. They extracted $x \ mod \ O(\beta)$, where β denotes the product of elements of smooth order. The attack is especially successful if $p-1$ has many small factors apart from one large factor of a 160 bits. The attack is explained below. Let g be a generator of order q and $O(\beta)$ denotes the order of β. If $z = \gamma^x \ mod \ p$ can be retrieved by attacking the protocol, then $x \ mod \ O(\gamma)$, where $\gamma = \prod \beta_i$(a product of distinct smooth order elements $\mod p$) can be obtained by using Pohlig-Hellman decomposition and finally the remaining part of x could be found from the public key y using Pollard-Lambda method.

In both the cases random primes are used for the computations and assumed to have many small factors apart from the large one for $p-1$. The difference is that, the former solved the problems with generators of order $|p-1|$ and the latter solved the problems with generators of order $|q|$ with computations restricted to the prime order subgroup q.

From the above discussion, the following problems are formulated and investigated.

1) Finding the exponent ($x \mod p-1$) in a specific range, i.e., $q \pm 2d$ in a group of order $|2q|$ with the order of y as $|2q|$ or $|q|$ and the order of g as $p-1$ (Type 1).

2) Finding the exponent ($x \mod q$) in a specific range, i.e., $\frac{q-1}{2} \pm d$ in a prime order subgroup of a safe prime or random prime with $p-1$ as 2ρ and one of the factor of 2ρ is large (q) with the order of g and y as q (Type 2).

3) Finding the exponent ($x \mod p-1$) with the assumption that $x \mod q$ lies near the middle of prime order subgroup, i.e., ($\frac{q-1}{2} \pm d$) in a random prime with $p-1$ as 2ρ, where ρ has many small factors along with a large q with the generator g is of order $p-1$ (Type 3).

3.1 Exponents in a Specific Range in a Group of Order $2q$

The use of a prime p of the form $p = 2q+1$ where q is also a prime is considered as safe. The safe primes preclude the Van Oorshot attack, since the partial Pohlig-Hellman decomposition yields only a single bit of information about the exponents x. The advantage of using safe primes is that, all group elements of Z_p^* other than ± 1 are known to have order either q or $2q$.

From Theorem 1, the set of smooth numbers over Z_p^* generates an inverse palindrome pattern. If the order of y is known, then the order of $-y$ may be decided by using this pattern.

1) y is of order $|2q|$.
In this section we present a method to solve the DLP, where the exponents x lies in the interval $[q \pm 2d]$, and d <100 bits. The method first converts y of order $2q$ into $-y$ and computes the logarithm of $-y$ in the prime order subgroup using the Pollard-Lambda method. The logarithm of y is obtained as follows.

Let $g^x = y$, where g and y are known, x to be solved and the order of g is $p-1$. The above equation can be rewritten as follows using Equations (5) and (6).

$$g_1^i = -y;$$

where g_1 is the generator of order q. Since,

$$\begin{aligned} g^x &= y \\ g^{q+2i} &= y \\ g^q.(g^2)^i &= y \\ (g^2)^i &= -y \\ g_1^i &= -y. \end{aligned}$$

Here i is assumed to lie in an interval [1, d]. Solve i by Pollard-Lambda method. Thus,

$$x = q + 2 * i \bmod 2q.$$

2) y is of order q.
When the order of y is q, the logarithm of y is calculated as follows. Let $g^x = y$. Since the order of y is q, it can be generated from g_1. Thereby, the equation can be rewritten as follows:

$$g_1^{x'} = y,$$

where g_1 is the generator of q and x' assumed to lie within the range $[(\frac{q-1}{2} \pm d]$. Now, the following equation is obtained by using Equations (5) and (6):

$$\begin{aligned} g^{x_1} &= -y, \\ g^{-1 \pm 2d} &= -y \\ g^{-1}g^{\pm 2d} &= -y \\ (g^{\pm 2})^d &= -y.g \\ g_1^d &= -y.g, \end{aligned}$$

where x_1 is $-1 \pm 2d$. Solve d using Pollard-Lambda method and solve x_1 using the equation $-1 \pm 2d$. Then,

$$\begin{aligned} x' &= (x_1 - p)/2 \bmod q \\ x &= 2x'. \end{aligned}$$

Since

$$\begin{aligned} g^x &= y \\ (g^2)^{x'} &= y \\ g_1^{x'} &= y. \end{aligned}$$

The equation for $-y$ is $g^{x_1} = -y$.

3.2 Exponent in a Specific Range in a Prime Order Subgroup of a Safe Prime of the Form $2q + 1$ or a Random Prime of $2\rho + 1$

In this section we present a method of solving DLP in a prime order subgroup of a safe prime and random prime of the form $2\rho + 1$ along with the assumption that the exponents lie in a specific range. In this type of primes, the set of smooth numbers over Z_p^* generates an inverse palindrome pattern for safe primes with respect to the factor base of elements of order q. For random primes the set is belongs to inverse palindrome pattern with respect to the factor base of elements of order ρ and to palindrome pattern with respect to the factor base of elements of order $2X$, where X is a prime or product of primes. If the order of y is known, then the order of $-y$ may be decided by using these patterns. The logarithm can be solved as follows.

1) Prime order subgroup of safe prime.
Let $g^x = y$, where g is the generator of prime order subgroup(q) of safe prime, y is of order q and x is assumed to lie within the range $[(\frac{q-1}{2} \pm d]$.

This equation can be rewritten as follows using Equations (5) and (6), which relates y and $-y$.

$$\begin{aligned} g_1^{x_1} &= -y \bmod 2q; \\ g^{x_1} &= -y^2 \bmod q. \end{aligned}$$

Since,

$$\begin{aligned} g^x &= y \\ g_1^{x_1} &= -y \\ g_1^{2x_1} &= -y^2 \\ (g_1^2)^{x_1} &= -y^2 \\ g^{x_1} &= -y^2, \end{aligned}$$

where x_1 is $-1 \pm 2d$, g_1 is the generator of order $2q$ and g is the generator of order q.

$$\begin{aligned} g^{x_1} &= -y^2; \\ g^{-1 \pm 2d} &= -y^2 \\ g^{-1}g^{\pm 2d} &= -y^2 \\ (g^{\pm 2})^d &= -y^2.g \\ g_x^d &= -y^2.g; \end{aligned}$$

The value for d can be obtain using Pollard-Lambda method and x_1 can be solved using the equation $-1 \pm 2d$. x can be solved as follows:

$$x = (x_1 - p)/2 \bmod q.$$

2) Prime order subgroup of random prime
Let $g^x = y$, where g is the generator of prime order subgroup(q) of random prime, y is of order q and x is assumed to lie within the range $[(\frac{q-1}{2} \pm d]$. The equation can be written as follows:

$$\begin{aligned} g^x &= y \mod q \\ g^{x_1} &= (-y)^2 \mod p. \end{aligned}$$

Since,

$$\begin{aligned} g^x &= y \mod q \\ g_2^{x_1} &= -y, \end{aligned}$$

where g_2 is the generator of order $2q$.

$$\begin{aligned} g_2^{2x_1} &= (-y)^2 \\ g^{x_1} &= (-y)^2, \end{aligned}$$

where x_1 is $-1 \pm 2d$.

$$\begin{aligned} g^{x_1} &= -y^2 \\ g^{-1 \pm 2d} &= -y^2 \\ g^{-1}g^{\pm 2d} &= -y^2 \\ (g^{\pm 2})^d &= -y^2.g \\ g_x^d &= -y^2.g; \end{aligned}$$

Solve d using Pollard-Lambda method and x_1 is solved using the equation $-1 \pm 2d$. Once the logarithm of x_1 is known, the equation $x_1 = q + 2x$ is to be solved to obtain the value of x.

On the other hand, the x_1 can be solved if the value $x_1 \ mod \ \gamma$ is leaked through the protocol design. The γ is the product of primes less than some small bound B, in such case, the computation needed for the adversary is relatively less. For example a random prime of size $\approx$ 1024 bits with factors of $p-1$ as many smooth factors along with a prime of size $\approx$ 160 bits is chosen for testing. Since one of the factors of $p-1$ is large of size 160 bits, the $x_1 \ mod \ q$ can not be solvable by using Pollard-Rho method. The Pollard-Rho method needs $O(\sqrt(2^{160}))$ group operation, which is $O(2^{80}))$ group operations. If $x_1 \mod \gamma$ is known, then the logarithm of remaining bits of x_1 is obtained as follows. Suppose $x_1 \ mod \ \gamma$ is ≈ 40 bits and $x_1 \ mod \ q$ is within the interval $[1, \ d]$ and ≈ 80 bits. Then the computation of remaining bits requires only $O(\sqrt(2^{40}))$ time and space using Pollard-Lambda method, which is a feasible computation power for the adversary.

3.3 Random Prime with $p-1=2\rho$

The random primes with factors of $p-1$ as many smooth factors along with a large prime and the order of the generator g as $p-1$, is considered for the exponents of specific range. The logarithm can be solved by using Pohlig-Hellman method. The method solves the logarithm $x \mod p$ in the subgroups by obtaining $x_i \ mod \ p_i$ and combining the results using the Chinese Remainder Theorem. The $x_i \ mod \ p_i$ can be solved by using Shanks or Pollard-Rho. Since one of the factor of $p-1$ is a large prime q, the $x_i \ mod \ q$ can not be solvable by using Pollard-Rho method. Alternate way to get the solution is, if $x_i \ mod \ \gamma$ is leaked through the protocol design by using Lim and Lee attack [7], then the logarithm of remaining bits of x_i is obtained by using Pollard-Lambda method.

Apart from the above two approaches, another approach to the above problem is, If the logarithm $x_i \ mod \ q$ is assumed to lie within the range $[(\frac{q-1}{2} \pm 1d]$, then it can be computed in $(O(\sqrt(d))$ group operations as follows:

$$\begin{aligned} g^x &= y \\ g^{\frac{p-1}{q}} &= y^{\frac{p-1}{q}} \\ g_1^{x_i} &= y_1 \mod q, \end{aligned}$$

where x_i assumed to lie in the range $[(\frac{q-1}{2} \pm d]$, g_1 is the generator of subgroup q and the reduction from g^x to $g_i^{x_i}$ is by using Pohlig-Hellman decomposition.

$$\begin{aligned} g_1^{x_i} &= y_1 \mod q \\ g_1^{x_1} &= (-y_1)^2. \end{aligned}$$

Since $g_1^{x_i} = y_1 \mod q$. This can be rewritten using the property which relates y and $-y$ as follows:

$$g^{x_1} = -y_1,$$

where g is the generator of order $|2q|$.

$$\begin{aligned} g^{2x_1} &= (-y_1)^2 \\ (g^2)^{x_1} &= (-y_1)^2 \\ g_1^{x_1} &= (-y_1)^2, \end{aligned}$$

where x_1 is $-1 \pm 2d$.

$$\begin{aligned} g_1^{x_1} &= -y_1^2 \\ g_1^{-1 \ \pm \ 2d} &= -y_1^2 \\ g_1^{-1} g^{\pm 2d} &= -y_1^2 \\ (g_1^{\pm 2})^d &= -y_1^2 . g_1 \\ g_x^d &= -y_1^2 . g_1; \end{aligned}$$

Once the logarithm of x_1 is known, the equation $x_1 = q + 2 * x_i$ is to be solved to obtain the value of $x_i \mod q$. The logarithm $x \bmod p$ can be solved using the Chinese Remainder Theorem on $x_i \bmod p_i$. The following section presents the results to support the claims made in the present section.

4 Experimental Results

In this section we describe the experimental results and give a representative selection of the problems solved. The purpose of our experiments is to produce the data on which we can base reliable statements about the expected running time of the proposed methods to solve the DLP. First we generated a database of approximately 100 problems in each type described below along with the necessary information to carry out the experiments:

- Problems of prime p with $p-1$ as $2q$ (safe prime) with the generator of order $|2q|$ and the order of y as $|2q|$, safe primes with the generator of order $|2q|$ and the order of y as $|q|$ for solving Section 3.1.
- Problems of safe primes with generator of order $|q|$, all computations are restricted within this prime order subgroup for Section 3.2.
- Problems of primes with $p-1$ as 2ρ, where ρ consist of small primes along with a big prime q with the generator of order $|q|$, and the computations are restricted within this prime order subgroup for Section 3.2.
- Problems of primes with $p-1$ as 2ρ, where ρ consist of small primes along with a big prime q with the generator of order $|2\rho|$ for Section 3.3.

A data file is produced with 6-tuple (p, d, g, y, t, F) with each tuple a sample problem.

p is the prime to be tested of size between 100 to 1024 bits.

d is the interval.

g is the generator, it may be a generator for a group or for a prime order subgroup.

y is g^x.

t is the type of the prime.

F is an array of factors of $p-1$.

Having built up the file the following algorithm is executed.

- Read a tuple (p, d, g, y, t, F);
- Find the type of the problem;
- Find the order of y using the factors of $p-1$;
- Use the methods discussed in the previous section according to the order of y and the type of the problem;
- Keep track of the computed run time;
- Repeat the above steps until all tuples are calculated.

4.1 Results on Selected List of Problems

This section presents the selected list of problems solved using the methods discussed in the previous section. In the following examples, p represents the prime, g represents the generator of the group $p-1$, $g1$ denotes the generator of the prime order subgroup and y is the element for which the DLP is solved. The experimental results are conducted on a Pentium 1V machine with 256MB RAM capacity.

For example, **Safe primes:**
y of order $2q$:
$p = 20000000000000000000000000000000000000$
00000000000000000000000000
00000000000000000000000000000000000000000
00000000000000000000000000
00000000000000000000000000000000000000000051447

g= 777

$y = 769367604420241674542275365954 98$
16469371006927217048384631749221265
30212347195976528377816908539301 2096
24526440458231666798854195467 20
97721872032646888627078423175416914084650 3947

the x solved is
$x = 10000000000000000000000000$
00000000000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000002199023281275

Random primes:
$p = 44232039047331017300449054 37$
20693717675480000000000000000000000000
00000000000000000000000000000000288016
92225669591915187401754372 97
14264388803

$g = 2$

$g1 = 30063376819308994735373472630510$
97661680341512777789431739830909931
30572137220797537127353192198583530 6
92480230514413880625807007 6319
949023406

$y = 11790709567996238855704542663179$
17272291084795000814271896986586616 5
05041456166021471206805642117680 0
74067817106447299701768142 07
3509942537983

$x = 22116019523665508650224527186034 6$
85883774000000000000000000000000000
00000000000000000000000023749813351
736362087846258956632 6
68224912910743076

Prime order subgroup of Random primes:
$p = 28721874389328679940809578569558 5$
67881693352764839668121224746091115 9
847224901457171

$g1 = 84356340542889230018958878775437 02$
01417341076475518645299815127171
96296200418796030

$y = 93487413609621210715861278866656915 5$
27237602371260435551747772482 73
3606227841586607

$x = 58022558939689821414242271998213062370$

The results are summarized in Tables 1, 2 and 3. Table 1 gives the average running time for solving DLP on different size problems of safe primes from 100 to 1024 bits. The variable d indicates the interval. Table 2 shows the average running time for prime order subgroup of safe primes and random primes.

Table 3 presents the running time for random prime with the generator of order $p-1$. The $p-1$ is assumed to have many small factors apart from the large one q and γ is the product of small factors. The DLP $x \mod p-1$ is to be obtained by computing $x_i \; mod \; \gamma$ and $x_i \; mod \; q$ and finally combining these results with Chinese Remainder Theorem. $x_i mod \gamma$ can be solved by using Pohlig-Hellman method. The $x_i \mod q$ is solved under two assumptions, when the γ bits of $x_i \mod q$ are not leaked and when they are leaked. In the former case the Lim and Lee attack is used and in the latter case only Pollard-Lambda is used. In all the above cases the logarithm in an interval [1, d] is solved using the Pollard-Lambda method. The average

Table 1: Running time of methods to solve DLP on safe primes

Problem size in bits	Safe prime with y as order $2q$				Safe prime with y as order q			
d	2^{10}	2^{20}	2^{30}	2^{40}	2^{10}	2^{20}	2^{30}	2^{40}
100	–	–	–	–	–	–	–	–
256	–	2	9m	4 h	–	1s	10m	4h 2m
512	–	3s	15m	5 h	–	2s	15m	4h 30m
1024	–	3s	1h 12m	13h 15m	–	3s	1h 30m	15 h

Table 2: Running time of methods to solve DLP on prime order subgroups

Problem size in bits	Prime order subgroup of safe prime				Prime order subgroup of random prime			
d	2^{10}	2^{20}	2^{30}	2^{40}	2^{10}	2^{20}	2^{30}	2^{40}
100	–	–	–	–	–	–	–	–
256	–	2s	11m	5 h	–	–	1s	9m
512	–	3s	16m	4h 30m	–	–	5s	13m
1024	–	2s	50m	15h	–	–	4s	4h

running time of the methods to solve the DLP reported in the Tables 1, 2 and 3 grows exponentially with respect to the order of the group or prime order subgroup and the interval d.

5 A Key Recovery Attack on Chang and Chang Password Key Exchange Protocol

This section briefly explains the key recovery attack on Chang and Chang novel three party key exchange protocol proposed by R.Padmavathy and Chakravarthy Bhagvati [13]. The key exchange protocols using passwords achieved great attention due to its simplicity and efficiency. On the other hand, the protocol should resist all types of password guessing attacks, since the password is of low entropy. Recently Chang and Chang proposed a novel three party simple key exchange protocol. They claimed the protocol is secure, efficient and practical. Unless their claims Yoon and Yoo presented an Undetectable online password guessing attack on the above protocol. In the similar line an another attack called as a key recovery attack on Chang and Chang protocol using the Undetectable online password guessing attack proposed by Yoon and Yon is proposed by R.Padmavathy and Chakravarthy Bhagvati [13]. The other improvements over password key exchange protocols are [2, 8, 15, 23]. The notations used in this protocol are listed below:

A, B**:** two communication parties.

S**:** the trusted server.

ID_A, ID_B, ID_S**:** the identities of A, B and S, respectively.

PW_A, PW_B**:** the passwords securely shared by A with S and B.

$E_{PW}(\cdot)$**:** a symmetric encryption scheme with a password PW.

r_A, r_B**:** the random numbers chosen by A and B, respectively.

p**:** a large prime.

g**:** a generator of order $p-1$.

R_A, R_B, R_S**:** the random exponents chosen by A, B and S, respectively.

N_A, N_B**:** $N_A = g^{R_A} \mod p$ and $N_B = g^{R_B} \mod p$.

$F_S(\cdot)$**:** the one-way trapdoor hash function (TDF) where only S knows the trapdoor.

$f_K(\cdot)$**:** the pseudo-random hash function (PRF) indexed by a key K.

K_{AS}, K_{AS}**:** a one time strong keys shared by A with S and B with S, respectively.

5.1 The Chang and Chang Password Key Exchange Protocol

The procedure followed in Chang-Chang [1] is given below:

Step 1. $A \rightarrow B$: $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A)\}$. User A chooses a random integer number r_A and a random exponent $R_A \in_R Z_p^*$, and then computes $N_A = g^{R_A}$ and $K_{AS} = N_A^{R_A}$. Then, A encrypts N_A by using his/her password PW_A like E_{PW_A}, (N_A) and computes two hash values $F_S(r_A)$ and $f_{K_{AS}}(N_A)$. Finally, A sends $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A)\}$ to B.

Step 2. $B \rightarrow S$: $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A), E_{PWB}(N_B), F_S(r_B), f_{K_{BS}}(N_B)\}$. User

Table 3: Running time of methods to solve DLP on random primes

Problem size in bits	Random prime with generator of order p-1				Random prime with generator of order p-1 and γ bits leaked			
d	2^{10}	2^{20}	2^{30}	2^{40}	2^{10}	2^{20}	2^{30}	2^{40}
100	–	–	–	–	–	–	–	
256	–	1s	14m	3h	–	–	14s	1h 10m
512	–	–	13m	4h	–	–	17s	2h 15m
1024	–	2s	57m	12h 30m	–	1s	35s	6h

B chooses a random integer r_B and a random exponent $R_B \in_R Z_p^*$, and then computes $N_B = g^{R_B}$ and $K_{AB} = N_B^{R_B}$. Then, B encrypts N_B by using his/her password PW_B like E_{PW_B}, (N_B) and computes two hash values $F_S(r_B)$ and $f_{K_{AB}}(N_B)$. Finally, B sends $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A), E_{PWB}(N_B), F_S(r_B), f_{K_{BS}}(N_B)\}$ to S.

Step 3. $S \rightarrow B$: $\{ N_B^{R_S}, f_{K_{AS}}(ID_A, ID_B, K_{AS}, N_B^{R_S}), N_A^{R_S}, f_{BS}(ID_A, ID_B, K_{BS}, N_A^{R_S})\}$. Server S decrypts $E_{PWA}(N_A)$ and $E_{PWB}(N_B)$ by using PW_A and PW_B to get N_A and N_B, respectively. Then, S gets r_A and r_B from $F_S(r_A)$ and $F_S(r_B)$ by using a trap door, respectively. To authenticate A and B, S computes $K_{AS} = N_A^{r_A}$ and $K_{BS} = N_B^{r_B}$ and then verifies $f_{K_{AS}}(N_A)$ and $f_{K_{BS}}(N_B)$, respectively. If successful, S chooses a random exponent $R_S \in_R Z_p^*$ and then computes, $N_A^{R_S}$ and, $N_B^{R_S}$, respectively. Finally, S computes two hash values $f_{K_{AS}}(ID_A, ID_B, K_{AS}, N_B^{R_S})$, $f_{K_{BS}}(ID_A, ID_B, K_{BS}, N_A^{R_S})$, and sends $\{ N_B^{R_S}, f_{K_{AS}}(ID_A, ID_B, K_{AS}, N_B^{R_S}), N_A^{R_S}, f_{BS}(ID_A, ID_B, K_{BS}, N_A^{R_S})\}$ to B.

Step 4. $B \rightarrow A$: $\{ N_B^{R_S}, f_{K_{AS}}(ID_A, ID_B, K_{AS}, N_B^{R_S}), f_K(ID_B, K)\}$. By using $K_{BS} = N_B^{r_B}$, B authenticates S by checking $f_{BS}(ID_A, ID_B, K_{BS}, N_A^{R_S})$. If successful, B computes the session key $K = (N_A^{R_S})_B^R = g_S^R R_A R_B$ and hash value $f_K(ID_B, K)$, and then sends $\{ N_B^{R_S}, f_{K_{AS}}(ID_A, ID_B, K_{AS}, N_B^{R_S}), f_K(ID_B, K)\}$ to A.

Step 5. $A \rightarrow B$: $\{f_K(ID_A, K)\}$ By using $K_{AS} = N_A^{r_A}$, A authenticates S by checking $f_{K_{AS}}(ID_A, ID_B, K_{AS}, N_B^{R_S})$. If successful A computes the session key $K = (N_B^{R_S})^{R_A} = g^{R_S R_A R_B}$, and authenticates B by checking $f_K(ID_B, K)$. If authenticates is passed, A computes and sends $f_K(ID_A, K)$.

Step 6. B authenticates A by checking $f_K(ID_A, K)$. If successful, B confirms A's knowledge of the session key $K = g^{R_S R_A R_B}$.

5.2 Undetectable Password Guessing Attacks on Chang-Chang Protocol

This section demonstrates the undetectable password guessing attack on Chang-Chang protocol. Assuming B as malicious party the procedure of the above attack is given below:

Step 1. $A \rightarrow B$: $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A)\}$.

Step 2. B records message $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A)\}$ from A.

Step 3. B guesses a password PWA' from password dictionary and gets N'_A.

Step 4. B chooses a random integer r_B and then computes $K_{BS} = N_A'^{r_B}$. Then, B encrypts N'_A by using his/her password PW_B like E_{PW_B}, (N'_A) and computes two hash values $F_S(r_B)$ and $f_{K_{BS}}(N'_A)$.

Step 5. $B \rightarrow S$: $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A), E_{PWB}(N'_A), F_S(r_B), f_{K_{BS}}(N'_A)\}$. B transmits $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A), E_{PWB}(N_B), F_S(r_B), f_{K_{BS}}(N_B)\}$.

Step 6. $S \rightarrow B$: $\{ N_A'^{R_S}, f_{K_{AS}}(ID_A, ID_B, K_{AS}, N_A'^{R_S}), N_A^{R_S}, f_{BS}(ID_A, ID_B, K_{BS}, N_A^{R_S})\}$. After receiving the message S can authenticate A and B by verifying $f_{K_{AS}}(N_A)$ and $f_{K_{BS}}(N'_A)$, respectively. S will compute $f_{K_{AS}}(ID_A, ID_B, K_{AS}, N_A'^{R_S})$ and $f_{BS}(ID_A, ID_B, K_{BS}, N_A^{R_S})$ to B.

Step 7. After receiving the message B simply compares $N_A'^{R_S} = N_A^{R_S}$. If $N_A'^{R_S} = N_A^{R_S}$, it follows that $PWA' = PWA$.

5.3 The Key Recovery Attack

A malicious party B guesses the password of A using Undetectable password guessing attack as proposed by Yoon and Yoo [22]. B uses the password of A for obtaining the session key between A and C, when A and C wants to communicate. The following procedure presents the attack in detail. In the present study, the prime p in the protocol is considered as safe prime and the exponents such as R_A and R_B are assumed as centered around the middle of the group or prime order subgroup and at the distance of $\approx 2^{40}$.

Step 1. $A \rightarrow C$: $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A)\}$. User A chooses a random integer number r_A and a random exponent $R_A \in_R Z_p^*$, and then computes $N_A = g^{R_A}$ and $K_{AS} = N_A^{R_A}$. Then, A

encrypts N_A by using his/her password PW_A like E_{PW_A}, (N_A) and computes two hash values $F_S(r_A)$ and $f_{K_{AS}}(N_A)$. Finally, A sends $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A)\}$ to C.

Step 2. B gets $\{ID_A, ID_B, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A)\}$ and from $E_{PWA}(N_A)$ decrypts N_A, since password is known and solves the ephemeral key R_A from N_A using the methods to solve the DLP for safe primes as discussed in Subsection 3.1.

Step 3. $C \rightarrow S$: $\{ID_A, ID_C, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A), E_{PWC}(N_C), F_S(r_C), f_{K_{CS}}(N_C)\}$. User C chooses a random integer r_C and a random exponent $R_C \in_R Z_p^*$, and then computes $N_C = g^{R_C}$ and $K_{CS} = N_C^{R_C}$. Then, C encrypts N_C by using his/her password PWC like E_{PWC}, (N_C) and computes two hash values $F_S(r_C)$ and $f_{K_{CS}}(N_C)$. Finally, C sends $\{ID_A, ID_C, ID_S, E_{PWA}(N_A), F_S(r_A), f_{K_{AS}}(N_A), E_{PWC}(N_C), F_S(r_C), f_{K_{CS}}(N_C)\}$ to S.

Step 4. $S \rightarrow C$: $\{ N_C^{R_S}, f_{K_{AS}}(ID_A, ID_C, K_{AS}, N_C^{R_S}), N_A^{R_S}, f_{CS}(ID_A, ID_C, K_{CS}, N_A^{R_S})\}$. Server S decrypts $E_{PWA}(N_A)$ and $E_{PWC}(N_C)$ by using PW_A and PW_C to get N_A and N_C, respectively. Then, S gets r_A and r_C from $F_S(r_A)$ and $F_S(r_C)$ by using a trap door, respectively. To authenticate A and B, S computes $K_{AS} = N_A^{r_A}$ and $K_{CS} = N_C^{r_C}$ and then verifies $f_{K_{AS}}(N_A)$ and $f_{K_{CS}}(N_C)$, respectively. If successful, S chooses a random exponent $R_S \in_R Z_p^*$ and then computes, $N_A^{R_S}$ and $N_C^{R_S}$, respectively. Finally, S computes two hash values $f_{K_{AS}}(ID_A, ID_C, K_{AS}, N_C^{R_S})$ $f_{K_{CS}}(ID_A, ID_C, K_{CS}, N_A^{R_S})$, and sends $\{ N_C^{R_S}, f_{K_{AS}}(ID_A, ID_C, K_{AS}, N_C^{R_S}), N_A^{R_S}, f_{CS}(ID_A, ID_C, K_{CS}, N_A^{R_S})\}$ to B.

Step 5. B gets $\{ N_C^{R_S}, f_{K_{AS}}(ID_A, ID_C, K_{AS}, N_C^{R_S}), N_A^{R_S}, f_{CS}(ID_A, ID_C, K_{CS}, N_A^{R_S}\}$ and from $N_C^{R_S}$ he computes the session key $(N_C^{R_S})^{R_A}$ is nothing but a session key between A and C $g^{R_A R_S R_C}$.

Step 6. $C \rightarrow A$: $\{ N_C^{R_S}, f_{K_{AS}}(ID_A, ID_C, K_{AS}, N_C^{R_S}), f_K(ID_C, K)\}$. By using $K_{CS} = N_C^{r_C}$, C authenticates S by checking $f_{CS}(ID_A, ID_C, K_{CS}, N_A^{R_S})$. If successful, C computes the session key $K = (N_A^{R_S})^{R_C} = g^{R_S R_A R_C}$ and hash value $f_K(ID_C, K)$, and then sends $\{ N_C^{R_S}, f_{K_{AS}}(ID_A, ID_C, K_{AS}, N_C^{R_S}), f_K(ID_C, K)\}$ to A.

Step 7. $A \rightarrow C$: $\{f_K(ID_A, K)\}$. By using $K_{AS} = N_A^{r_A}$, A authenticates S by checking $f_{K_{AS}}(ID_A, ID_C, K_{AS}, N_C^{R_S})$. If successful A computes the session key $K = (N_C^{R_S})^{R_A} = g^{R_S R_A R_B}$, and authenticates C by checking $f_K(ID_C, K)$. If authenticates is passed, A computes and sends $f_K(ID_A, K)$.

Step 8. C authenticates A by checking $f_K(ID_A, K)$. If successful, C confirms A's knowledge of the session key $K = g^{R_S R_A R_C}$.

5.4 Experimental Results

From the discussion of key recovery attack on Chang-Chang protocol, it is observed that the computation of R_A, the ephemeral key from N_A is the key issue. This leads to recover the session key by the malicious party B. The ephemeral keys may be unique for each session or they may be reused for different sessions of a same party. For example, the ANSI X9.42 standard, which specifies several Diffie-Hellman protocols states that an ephemeral key is a "private or public key that is unique for each execution of a cryptographic schemes". Other protocols do not place any restrictions on the reuse of ephemeral keys [10]. In Chang and Chang protocol the ephemeral keys are unique for each session. One way of retrieving these keys are to solve the mathematical hard problem such as Discrete Logarithm Problem (DLP).

The ephemeral keys are solved by using Pohlig-Hellman method in [13]. The ephemeral keys are dynamic and changes for every session between Alice and Bob while the static keys remains the same and lives longer. Since the life time of ephemeral keys are short, it is hard to recover these keys within the short span of time by using the attacks with the target of solving the DLP. Hence, the keys are assumed to be around the middle of the group or prime order subgroup and at the distance of 2^{40}. The Type-1 problems as discussed in Section 3.1, i.e., safe primes are chosen for testing.

Experiments are conducted based on the above characteristics. Let us describe clearly, first we generated a database of approximately 100 problems of safe primes along with the necessary information to carry out the experiments. Having built up the data base the key recovery attack is implemented on Chang and Chang password key exchange protocol. The R_A from N_A is solved by using the methods discussed in Section 3.1 for safe primes. Table 4 shows the average running time to mount the attack on the Chang and Chang password key exchange protocol.

6 Conclusion

In the present study we developed algorithms and presented running times for the following problems.

1) Finding the exponent $(x \mod p-1)$ in a specific range, i.e., $q \pm 2d$ in a group of order $|2q|$ with the order of y as $|2q|$ or $|q|$ and the order of g as $p-1$ (Type 1).

2) Finding the exponent $(x \mod q)$ in a specific range, i.e., $\frac{q-1}{2} \pm d$ in a prime order subgroup of a safe prime or random prime with $p-1$ as 2ρ and one of the factor of 2ρ is large (q) with the order of g and y as q (Type 2).

3) Finding the exponent $(x \mod p-1)$ with the assumption that $x \mod q$ lies near the middle of prime

Table 4: The average running time to mount the attack

Problem size in bits	Running time
256	5hr
512	6hr
1024	16hr

order subgroup, i.e., $(\frac{q-1}{2} \pm d)$ in a random prime with $p-1$ as 2ρ, where ρ has many small factors along with a large q with the generator g is of order $p-1$ (Type 3).

In the literature, it is reported that short exponents are vulnerable. Our algorithms show that exponents near the middle of a group or a prime order subgroup are also vulnerable. In addition, for Type 3 problems we extended the Pohlig-Hellman with Lim-Lee attack and our new method. Our analysis will help the cryptosystems to generate safe keys. This leads to avoid the keys that are vulnerable to the methods reported in the present study. The properties and the investigation reported in the present study are on specific instances of prime fields with $p-1$ as 2ρ. The work can be extended for the primes of the form $2^n\rho+1$. Also, the key recovery attack is mounted on Chang and Chang password key exchange protocol using the methods proposed in the present study to solve the DLP for ephemeral keys.

References

[1] C. C. Chang and Y. F. Chang, "A novel three party key exchange protocol," *Computer Standards and Interfaces*, vol. 26, no. 5, pp. 471–476, 2004.

[2] T. Y. Chang, M. S. Hwang, and W. P. Yang, "A communication-efficient three party password authenticated key exchange protocol," *Information Sciences*, vol. 181, pp. 217–226, 2011.

[3] D. Coppersmith, A. M. Odlyzko, and R. Schroeppel, "Discrete logarithms in GF(p)," *Algorithmica*, 1986.

[4] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Information Security*, vol. 22, no. 6, pp. 644–654, 1976.

[5] T. ElGamal, "A public key cryptosystem and a signature based on discrete logarithms," *IEEE Information Theory*, vol. IT-31, no. 4, pp. 469–472, 1985.

[6] D. E. Knuth, *The Art of computer programming.* Addisen Wesley, 1973.

[7] C. H Lim and P. J. Lee, "A key recovery attack on discrete log-based schemes using prime order subgroup," LNCS 1294, Springer-Verlag, pp. 249–263, 1997.

[8] J. W. Lo, J. Z. Lee, M. S. Hwang, and Y. P. Chu, "An advanced password authenticated key exchange protocol for imbalanced wireless networks," *Journal of Internet Technology*, vol. 11, no. 7, pp. 997–1004, 2010.

[9] K. S. McCurely, "The discrete logarithm problem," in *Proceedings of Symposia in Applied Mathematics*, vol. 42, pp. 49–74, 1990.

[10] A. Menezes and U. Berkant, "On reusing ephemeral keys in diffie-hellman key agreement protocols," *International Journal of Applied Cryptography*, vol. 2, no. 2, pp. 154–161, 2010.

[11] P. C. Van Oorschot and M. J. Wiener, "On diffie-hellman key agreement with short exponents," in *Advances in Cryptology (EUROCRYPT'96)*, LNCS 1070, pp. 332–343, 1996.

[12] R. Padmavathy and C. Bhagvati, "Ephemeral key recovery using index calculus method," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 13, no. 1, pp. 29–43, 2009.

[13] R. Padmavathy and C. Bhagvati, "A key recovery attack on chang and chang password key exchange protocol," in *International Conference on Computer and Network Technology*, pp. 176–181, 2009.

[14] R. Padmavathy and C. Bhagvati, "Methods to solve discrete logarithm problem for ephemeral keys," in *International Conference on ARTCOM*, pp. 704–708, 2009.

[15] H. K. Pathak and M. Sanghi, "Simple three party key exchange protocol via twin diffie-hellman problem," *International Journal of Network Security*, vol.15, no.4, pp. 256–264, 2013.

[16] S. Pohlig and M. Hellman, "An improved algorithm fro computing logarithms over gf(p) and its cryptographic significance," *IEEE Transactions on Information Theory*, vol. 24, pp. 106–110, 1978.

[17] J. M. Pollard, "Monte carlo methods for index computation (mod p)," *Mathematics of Computation*, vol. 32, no. 143, pp. 106–110, 1978.

[18] C. Studholme, "Discrete logarithm problem," *Research paper requirement (milestone) of the PhD program at the University of Toronto*, 2002.

[19] E. Thome, "Computation of discrete logarithms in f_2^{607}," in *Advances in Cryptology (ASIACRYPT'01)*, LNCS 2248, pp. 107–124, 2001.

[20] D. Weber, "Computing discrete logarithms with the general number field sieve," in *Algorithmic Number Theory*, LNCS 1122, pp. 391–403, 1996.

[21] D. Weber and T.Denny, "The solution of mccurleys discrete log challenge," in *Advances in Cryptology (CRYPTO'98)*, LNCS 1462, pp. 458–471, 1998.

[22] E. J. Yoon and K. Y. Yoo, "Improving the novel three-party encrypted key exchange protocol," *Computer Standards and Interfaces*, vol. 30, pp. 309–314, 1996.

[23] Y. Zeng, J. Ma, and S. Moon, "An improvement on a three party password based key exchange protocol using weil pairing," *International Journal of Network Security*, vol. 11, no. 1, pp. 17–22, 2010.

R. Padmavathy received an M.tech degree from Andhra University and a Ph.D from the University of Hyderabad, India. At present she is working as a faculty member at the National Institute of Technology,

Warangal. Her research interests include Information security, Cryptology and Network Security.

Chakravarthy Bhagvati received his Ph.D deree from RPI Newyork, USA. At present he is a professor at University of Hyderabad, Hyderabad, India. He published a number of papers in International Journals. His research interests include Image Processing, Computer Vision, Pattern Recognition, OCR for telugu and Cryptography.

A Fully Secure Attribute Based Broadcast Encryption Scheme

Qinyi Li and Fengli Zhang
(Corresponding author: Qinyi Li)

School of Computer Science and Engineering, University of Electronic Science and Technology of China
No.2006, Xiyuan Avenue, West Hi-tech Zone, Chengdu, 611731, P.R.China
(Email: leebluedreams@gmail.com)
(Received Nov. 12, 2012; revised and accepted Nov. 06, 2013)

Abstract

Besides traditional functionality of broadcast encryption, attribute based broadcast encryption (ABBE) provides more flexible assess control mechanism over encrypted data. In this paper, a ciphertext-policy ABBE (CP-ABBE) scheme is proposed. In such scheme, user's private key is attached with attributes and an index while ciphertext is associated with an access structure and a broadcast set. A key will decrypt a ciphertext if and only if its attributes satisfy the access policy and its index is included in the broadcast set. Our construction is based on composite order bilinear group and its security can be reduced to three static intractable assumptions through famous dual system encryption methodology in fully secure model. Some application aspects of our ABBE scheme and comparisons between such scheme and existing ABBE schemes are also provided.

Keywords: Access control, attribute based broadcast encryption, broadcast encryption, ciphertext policy, fully secure

1 Introduction

In broadcast encryption (BE) [4, 7, 14], there are n users who possess the decryption key issued by a dealer. A broadcaster may distribute confidential content for an intended recipient set $S \subset \{1, 2, \cdots, n\}$ who listens over the broadcast channel. Only the ones in S can decrypt the ciphertext while others gain no decryption privilege. Various kind of BE have been proposed such as identity based BE [6, 18], anonymous BE [2], and broadcast authentication scheme [16].

Attribute based encryption (ABE), first formalized in [17] enables expressive access policy over encrypted data. There are two flavors of ABE, ciphertext-policy ABE (CP-ABE)and key-policy ABE (KP-ABE). In CP-ABE, sender specifies an access structure in ciphertext which defines over universal attributes set. User's key is associated with some attributes. A key can decrypt a ciphertext if and only if its attributes satisfy the access structure. In KP-ABE, the situation is inversed. That is key is attached with access structure and ciphertext is corresbonded to attributes. CP-ABE can be used to realize secure cloud storage while KP-ABE is a useful tool for pay-TV system. For example, in a cloud storage system of an university, a user encrypts the file with an access policy $\mathbb{A} = \{(\text{“Title = Professor”}\ \mathsf{AND}\ \text{“Institute = CS”})\ \mathsf{OR}\ \text{“Positions = Dean”}\}$ and stores it to sever. Then all the professors in institute of computer science or all the deans in this university may login the server and get file's content by using their corresponding attribute key.

Attribute based broadcast encryption (ABBE) has richer structure and enjoys more flexible functionality than traditional BE schemes. In ABBE scheme, user's key (i.e. decrypt privilege) is not only depends on his index but also attributes. If these two factors satisfy the requirements of broadcast set and some access policy, a key can be employed to decrypt the ciphertext. A natural functionality of ABBE is to realize direct revocation in ABE sense. By employing ABBE scheme, revoked users' identities will not be added to recipient set and weather they satisfy the access policy they cannot decrypt the ciphertext. Traditional CP-ABE scheme which supports negative attribute expression is able to realize direct revocation by adding the revoked user's identity (or index) to access formula conjunctively in negative form. Here an identity is treated as an attribute. However, there is only one such CP-ABE scheme in the literatures belonging to [15], furthermore, its security can be proved merely in the ideal model called generic group model. Due to the different access control levels of ABBE from BE and ABE framework, we describe another application for ABBE. Consider the following example: there are a number of students belonging to different schools in an university. All of them are described by some attributes such as gender, grade and nationality. It is feasible to manage these students under each school while schools are supervised by manager of the university. That is students' attribute

keys are issued by school manager and school managers' keys are distributed by university. For this hierarchical structure, school and attribute entity can be controlled by BE mechanism and access formula, respectively. Suppose some notifications will be distributed to some students in school of mathematics and school of computer science who satisfy the condition $\mathbb{A}$ = {("GENDER = FEMALE" OR "GRADE = 5") AND "NATIONALITY = FRANCE"}. Note, since students' keys are not issued by the same school, thus traditional ABE scheme cannot provide proper solution. By using ABBE scheme with ciphertext policy, sender will specify these two schools as index and access condition $\mathbb{A}$ in the ciphertext. We will show how to use our ABBE scheme to provide an concrete and efficient solution in the latter section.

Full security model is seen as a "right model" witch reflects the real world property. In such model, adversary may choose its target adaptively. A weak and unrealistic model is "selective security model" where the adversary must declare its attack target before the system is set up. That is in the real world, the provably selective secure schemes are actually secure only if adversary is particularly interested in few "important" targets thus they are insecure in general cases. The formal definition of fully secure model is provided in following section.

Related Work. In [13], the authors proposed a novel ABBE scheme which supports restricted access formula (AND, NOT expression) and enjoys low decryption cost. In such scheme, both key's attribute and broadcast ciphertext's access policy are expressed as attribute polynomials. All the intended receivers will get the same greatest common polynomial from their own polynomial and the ciphertext's polynomial so as to proceed decryption. Other users are not able to get such polynomial. The security of this scheme can only be argued in the weak and ideal model called generic group model. Attrapadung *et al.* [1] presented another kind of ABBE schemes. They combine two state of the art ABE schemes with BGW broadcast encryption scheme [4] and Lewko *et al.*'s revocation encryption scheme [11] conjunctively to get four ABBE schemes (two key-policy schemes and two ciphertext-policy schemes). However, both of these above schemes can only archieve selective security. Junod and Karlov (2010) provided another ABBE scheme which supports general conjunctive normal form (CNF) and disjunctive normal form (DNF) Boolean formula. Its security also be proved in generic group model. A natural demerit of general attribute based schemes is the size of ciphertext which grows linearly with the attributes used. Based on BGW broadcast encryption scheme, Zhou and Huang presented an efficient and selectively secure ABBE scheme with constant-size ciphertext [19]. In the scheme, sender can specify both actual recipients and access policy in ciphertext but only restricted AND policy is supported. Its security is based on n-DBDHE assumption.

Our Contribution. In this paper, we propose a fully secure ciphertext-policy ABBE (CP-ABBE) scheme which supports any monotone access structure. In our scheme, user's key is associated with index and attributes, an access structure and broadcast set is embedded in ciphertext. The decryption can be done if the condition on attributes on the ABE part holds as usual and, in addition, user's index inside the broadcast. We utilize the current fully secure CP-ABE scheme by [10] and semi-static secure BE scheme proposed in [8] as building blocks. The main obstacle of our work is that such blocks are constructed in different algebraic structure, i.e. prime and composite order bilinear group respectively, and their security are based on different assumptions. We prove the security of our scheme through *dual system encryption methodology* (DSEM) in composite order bilinear group. To the best of our knowledge, such ABBE scheme is the first one which achieves full security.

Oganization. The rest of this paper is organized as follows. In Section 2, a brief review of some background of composite order bilinear group and relative assumptions used in our scheme is presented. In addition, the formal definitions of CP-ABBE algorithm and its security is given. The proposed scheme is presented in Section 3. The security of our scheme is analyzed in Section 4. Finally, some discussions and conclusions are provided in Sections 5 and 6.

2 Preliminary

In this section, background information concerning access structure, linear secret share scheme (LSSS), composite order bilinear group and relative intractable complexity assumptions is provided. After that, the definition of CP-ABBE scheme and its full security model are given. Here we address some notations which will be used latter. Let $\mathcal{X}$ be a set and $x \leftarrow_R \mathcal{X}$ denotes the operation of picking an element x uniformly at random from $\mathcal{X}$. Let $\mathbb{N}$ denote the set of natural numbers. We say a function $f(\cdot)$ is negligible for security parameter $\lambda \in \mathbb{N}$, if for every $c > 0$ there exists a λ_c such that $f(\lambda) < 1/\lambda^c$ holds for all $\lambda > \lambda_c$.

2.1 Access Structure

Definition 1. *(Access structure [3])*
Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}}$ is monotone if $\forall B, C$:if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$, i.e $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}} \backslash \{\phi\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.

Definition 2. *(Linear Secret-Sharing Schemes (LSSS) [10])*

A secret-sharing scheme Π *over a set of parties* $\mathcal{P}$ *is called linear (over* $\mathbb{Z}_N$*) if*

1) The shares for each party form a vector over $\mathbb{Z}_p$.

2) There exists a matrix A with ℓ rows and n columns called the share-generating matrix for Π. For all $x = 1, 2, \ldots, \ell$, the x'th row of A is labeled by a party $\rho(x)$ (ρ is a function from $1, 2, \ldots, \ell$ to parties $\mathcal{P}$. When we consider the column vector $\vec{v} = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $A\vec{v}$ is the vector of ℓ shares of the secret s according to Π. The share $(Av)_x$ belongs to the party $\rho(x)$.

Every LSSS has a very important property that is linear reconstruction *property defined as follow: Suppose that* Π *is an LSSS for the access structure* $\mathbb{A}$*. Let* $\omega \in \mathbb{A}$ *be any authorized set and let* $I \subset \{1, \cdots, \ell\}$ *be defined as* $I = \{x : \rho(x) \in \omega\}$*. Then there exist constants* $\{\mu_x \in \mathbb{Z}_N\}_{x \in I}$ *such that if* $\{\lambda_x\}$ *are valid shares of any secret* s *according to* Π*, then* $\sum_{x \in I} \mu_x \lambda_x = s$.

In this paper, we consider the LSSS over $\mathbb{Z}_N$ where $N = p_1 p_2 p_3$ for distinct primes p_1, p_2 and p_3. The definition is almost the same as above. Meanwhile, we only consider the monotone access structure. Since the equivalence between any monotone access structure and LSSS has been proved in [3], we ill use LSSS to express access structure. As in [10] we do restrict the function $\rho(x)$ as a bijection.

2.2 Composite Order Bilinear Groups

The composite order groups equipped with bilinear map were first introduced in [5]. Let an group parameters generator algorithm $\mathcal{G}$ take as input the security parameter λ and outputs the parameters of bilinear group $\mathbb{G} = (N = p_1 p_2 p_3, g, G, G_T, \hat{e})$ where p_1, p_2 and p_3 are distinct primes. G and G_T are cyclic groups of order $N = p_1 p_2 p_3$ multiplicative group action. $\hat{e} : G \times G \to G_T$ is bilinear map with following properties:

- Bilinear: $\forall g, h \in G$ and $a, b \in \mathbb{Z}_N$, $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$;
- Non-degenerate: $\exists g \in G$ such that $\hat{e}(g, g)$ has order N in G_T.

We further require that the group operations in G and G_T as well as the bilinear map $\hat{e}$ are computable in polynomial time with respect to λ. We use G_{p_1}, G_{p_2} and G_{p_3} to denote the subgroup of G with order p_1, p_2 and p_3 respectively. Here, we note $G = G_{p_1} \times G_{p_2} \times G_{p_3}$ and element in G can be expressed by the product of three unique elements in G_{p_1}, G_{p_2} and G_{p_3}. For example, any $T \in G$ can be expressed by $T = h_1 h_2 h_3$ uniquely where $h_1 \in G_{p_1}$, $h_2 \in G_{p_2}$ and $h_3 \in G_{p_3}$. We say "h_1 is the G_{p_1} part of T", "h_2 is the G_{p_2} part of T" and "h_3 is the G_{p_3} part of T".

Composite order bilinear group enjoys the orthogonality property that is for any two elements h_i and h_j from distinct subgroups G_{p_i} and G_{p_j} where $i \neq j$, we have $\hat{e}(h_i, h_j) = 1$. This property is crucial in our concrete construction and its security proof.

2.3 Computational Assumptions

The security of our CP-ABBE schemes is based on the the intractability of the following three static number theoretical assumptions which are used in [12]. The generic complexity of such assumptions are also proved in [12].

Assumption 1. *Given a group parameters generator* $\mathcal{G}$*, we define the following distribution:* $\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, \hat{e}) \leftarrow_R \mathcal{G}$, $g \leftarrow_R G_{p_1}, X_3 \leftarrow_R G_{p_3}$, $D = (\mathbb{G}, g, X_3)$, $T_1 \leftarrow_R G_{p_1 p_2}, T_2 \leftarrow_R G_{p_1}$*. We define the advantage of an algorithm* $\mathcal{A}$ *in breaking Assumption 1 to be:* $Adv^1_{\mathcal{G},\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$.

Definition 3. *We say that* $\mathcal{G}$ *satisfies Assumption 1 if* $Adv^1_{\mathcal{G},\mathcal{A}}(\lambda)$ *is a negligible function of* λ *for any polynomial time algorithm* $\mathcal{A}$.

Assumption 2. *Given a group parameters generator* $\mathcal{G}$*, we define the following distribution:* $\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, \hat{e}) \leftarrow_R \mathcal{G}$, $g, X_1 \leftarrow_R G_{p_1}, X_2, Y_2 \leftarrow_R G_{p_2}, X_3, Y_3 \leftarrow_R G_{p_3}$, $D = (\mathbb{G}, g, X_1 X_2, X_3, Y_2 Y_3)$, $T_1 \leftarrow_R G, T_2 \leftarrow_R G_{p_1 p_3}$*. We define the advantage of an algorithm* $\mathcal{A}$ *in breaking Assumption 2 to be:* $Adv^2_{\mathcal{G},\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$.

Definition 4. *We say that* $\mathcal{G}$ *satisfies Assumption 2 if* $Adv^2_{\mathcal{G},\mathcal{A}}(\lambda)$ *is a negligible function of* λ *for any polynomial time algorithm* $\mathcal{A}$.

Assumption 3. *Given a group parameters generator* $\mathcal{G}$*, we define the following distribution:* $\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, \hat{e}) \leftarrow_R \mathcal{G}, \alpha, s \leftarrow_R \mathbb{Z}_N$, $g \leftarrow_R G_{p_1}, X_2, Y_2, Z_2 \leftarrow_R G_{p_2}, X_3 \leftarrow_R G_{p_3}$, $D = (\mathbb{G}, g, g^\alpha X_2, X_3, g^s Y_2, Z_2)$, $T_1 = \hat{e}(g, g)^{\alpha s}, T_2 \leftarrow_R G_T$*. We define the advantage of an algorithm* $\mathcal{A}$ *in breaking Assumption 3 to be:* $Adv^3_{\mathcal{G},\mathcal{A}} = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$.

Definition 5. *We say that* $\mathcal{G}$ *satisfies Assumption 3 if* $Adv^3_{\mathcal{G},\mathcal{A}}(\lambda)$ *is a negligible function of* λ *for any polynomial time algorithm* $\mathcal{A}$.

2.4 Definitions of CP-ABBE Scheme and Its Security Model

Let $\mathcal{U}$ denote the set of index, $\mathcal{N}$ denote the universal attribute set and $\boldsymbol{AS}$ be an access structure family over $\mathcal{N}$. Here we describe the CP-ABBE scheme in key encapsuled manner and we can use some symmetric encryption scheme to encrypt or decrypt the actual messages under such key. We assume such key space is $\mathcal{K}$ A CP-ABBE scheme consists of four PPT algorithms and an additional key delegation algorithm.

1) **Setup**(1^λ)**.** The set up algorithm takes as input the security parameter λ and outputs the public parameters PK and master secret key MSK.

2) **KeyGen**(MSK, i, ω)**.** The key generation algorithm takes the user's index $i \in \mathcal{U}$, an key attributes $\omega \subseteq \mathcal{N}$ and MSK as input and outputs the secret key $SK_{(i,\omega)}$.

3) **Encrypt**$(PK, S, \mathbb{A})$**.** The encryption algorithm takes as input the public parameters PK, a broadcast index list $S \subseteq \mathcal{U}$ and an access structure $\mathbb{A} \in \boldsymbol{AS}$. It outputs a pair $\langle Hdr, K\rangle$ where Hdr is called the *header* and $K \in \mathcal{K}$ is a message encryption key.

4) **Decrypt**$(S, i, SK_{(i,\omega)}, Hdr)$**.** The decryption algorithm takes as input a secret key $SK_{(i,\omega)}$ as well as a ciphertext CT. If $i \in S$ and ω satisfies the access structure $\mathbb{A}$ (denoted by $\omega \in \mathbb{A}$) associated with the Hdr, decryption algorithm can recover key K. Otherwise, if $ID \notin S$ or $\omega \notin \mathbb{A}$, the decryption algorithm outputs special symbol $\perp$.

5) **Delegate**$(SK_{(i,\omega)}, \omega')$**.** The delegation algorithm takes as input a private key for index i and attribute set ω, a restricted attribute set ω' s.t. $\omega' \subseteq \omega$. It then outputs a new secret key $SK_{(i,\omega')}$.

We now describe the IND-CPA (indistinguishability against chosen plaintext attack) definition of CP-ABBE scheme in the fully secure model. The formal secure game between adversary $\mathcal{A}$ and challenger $\mathcal{B}$ is as follows:

1) **Setup.** Assume universal attribute set $\mathcal{N}$ and user set $\mathcal{U}$ as well as family $\boldsymbol{AS}$ is pre-defined. The challenger $\mathcal{B}$ runs set up algorithm to get the public parameters PK and master-key MSK. It then passes PK to adversary $\mathcal{A}$.

2) **Phase 1.** The adversary $\mathcal{A}$ adaptively issues private key queries for index $i \in \mathcal{U}$ and key attribute $\omega \subset \mathcal{N}$. The challenger $\mathcal{B}$ runs key generation algorithm to generate the corresponding private keys $SK_{(i,\omega)}$ and gives them to $\mathcal{A}$.

3) **Challenge.** In this phase, $\mathcal{A}$ submits a challenge broadcast set S^* and access structure $\mathbb{A}^*$ such that for all index i and attribute set ω queried in **Phase 1**, we have $i \notin S^*$ and $\omega \notin \mathbb{A}^*$. Then $\mathcal{B}$ runs **Encrypt**$(PK, S^*, \mathbb{A}^*)$ to get $\langle Hdr^*, K_0\rangle$ and randomly chooses $K_1 \leftarrow_R \mathcal{K}$. It flips a coin $b \in \{0, 1\}$ and gives $\langle Hdr^*, K_b\rangle$ to $\mathcal{A}$.

4) **Phase 2.** In this phase, $\mathcal{B}$ acts almost the same as in **Phase 1** except it is unable to ask key for attribute ω and index i such that $\omega \in \mathbb{A}^*$ and $i \in S^*$.

5) **Guess.** $\mathcal{A}$ outputs the guess bit b' for b and successes if $b' = b$.

We define adversary $\mathcal{A}$'s advantage in above game for security parameter λ with: $Adv_{\mathcal{A}}^{CP-ABBE}(\lambda) = |\Pr[b' = b] - 1/2]|$.

Definition 6. *We say CP-ABBE scheme is IND-CPA secure if, for all broadcast set $S \in \mathcal{U}$ and access structure $\mathbb{A} \in \boldsymbol{AS}$, no polynomial-time adversary can win the above game with non-negligible advantage with respect to security parameter λ.*

In contrast to above definition, the weak selective security notion is defined similarly as above game, except that adversary must submit the challenge broadcast set S^* and ciphertext access structure $\mathbb{A}^*$ before it sees the public parameters.

3 Fully Secure CP-ABBE Scheme

In this section, we give our construction of CP-ABBE scheme.

Setup. The set up algorithm takes as input a security parameter λ, universal attribute set $\mathcal{N}$ and user set $\mathcal{U}$ where $|\mathcal{U}| = n$. It runs the group generator $\mathcal{G}$ to get the group description $\mathbb{G} = (N = p_1p_2p_3, g, G, G_T, \hat{e})$. Then it chooses $g, h, u_1, u_2, \cdots, u_n \in G_{p_1}$ and $a, \alpha \in \mathbb{Z}_N$. For each attribute $i \in \mathcal{N}$, the set up algorithm chooses $t_i \in \mathbb{Z}_N$ and computes $T_i = g^{t_i}$. The public parameters PK are published as:

$$(N, \hat{e}(g, g)^\alpha, g, g^a, \{u_j\}_{j\in\mathcal{U}}, \{T_i\}_{i\in\mathcal{N}}).$$

The master secret key is g^α and G_{p_3}'s random generator X_3.

KeyGen. The key generation algorithm takes an user index $k \in \mathcal{U}$, an attributes set $\omega \subseteq \mathcal{N}$ and PK as inputs. It then chooses $r \in \mathbb{Z}_N$ and $R_3, R_3', R_{3,i}, R_{3,j} \in G_{p_3}$ randomly. The secrete key for user index k and attribute set ω is as follow:

$$\begin{aligned} K^{(1)} &= g^{\alpha+ar}u_k^r R_3, \quad K^{(2)} = g^r R_3', \\ \{K_j^{(3)} &= {u_j}^r R_{3,j}\}_{j\in\mathcal{U}\setminus\{k\}}, \quad \{K_i^{(4)} = T_i^r R_{3,i}\}_{i\in\omega}. \end{aligned}$$

Encrypt. The encryption algorithm takes as input a message $M \in G_T$, an access structure $\mathbb{A}(A, \rho)$ (we assume that A is a $\ell \times n$ matrix and ρ is map from each row x of A) to an attribute $\rho(x)$), public parameters PK and a broadcast set $S \subseteq \mathcal{U}$. It chooses a random $s \in \mathbb{Z}_N$. A secret share vector $\vec{v}$ is also chosen as $(s, v_2, v_3, \cdots, v_n) \in \mathbb{Z}_N^n$ and for every row of A (denoted by A_x), sender gets secret shares as $\lambda_x = A_x\vec{v}$ for all x and chooses $r_x \in \mathbb{Z}_N$. The symmetric key and header is formed as:

$$\begin{aligned} K &= \hat{e}(g, g)^{\alpha s}, \\ \{C_x^{(1)} &= g^{a\lambda_x}T_{\rho(x)}^{-r_x}, \\ C_x^{(2)} &= g^{r_x}\}_{\forall x}, \\ C^{(3)} &= (\textstyle\prod_{j\in S} u_j)^s, \quad C^{(4)} = g^s. \end{aligned}$$

Here, K is used to encrypt data in some symmetric encryption scheme and header Hdr =

$\left(\{C_x^{(1)}, C_x^{(2)}\}_{\forall x}, C^{(3)}, C^{(4)}\right)$.

Decrypt. The decryption algorithm takes a secret key for index k and attributes set ω, a header Hdr for access structure $\mathbb{A}(A,\rho)$ and broadcast set $S \subset \mathcal{U}$. If $k \in S$ and $\omega \in \mathbb{A}(A,\rho)$, then there must exist the reconstruction coefficients $\{\mu_x\}$ satisfies $\sum_{x\in I}\mu_x\lambda_x = s$. The decryption algorithm proceeds as:

$$\begin{aligned}
& \hat{e}\left(K^{(1)}\prod\nolimits_{j\in S\setminus\{k\}} K_j^{(3)}, C^{(4)}\right) \qquad (1)\\
= \;& \hat{e}\left(g^{\alpha+ar}u_k^r\prod\nolimits_{j\in S\setminus\{k\}} u_j^r R_{3,j}, g^s\right)\\
= \;& \hat{e}(g^\alpha, g^s)\cdot\hat{e}(g^{ar}, g^s)\cdot\hat{e}(\prod\nolimits_{j\in S\setminus\{k\}} u_j^r, g^s)\\
= \;& \hat{e}(g,g)^{\alpha s}\cdot\hat{e}(g,g)^{ars}\cdot\hat{e}(g,\prod\nolimits_{j\in S} u_j)^{rs}\\
= \;& A_1.
\end{aligned}$$

$$\begin{aligned}
\hat{e}(K^{(2)}, C^{(3)}) &= \hat{e}\left(g^r R_3', (\prod\nolimits_{j\in S} u_j)^s\right) \qquad (2)\\
&= \hat{e}\left(g, \prod\nolimits_{j\in S} u_j\right)^{rs}\\
&= A_2.
\end{aligned}$$

For all $x \in I$ and $\rho(x) = i \in \omega$, we have:

$$\begin{aligned}
& \prod_{x\in I}\left(\hat{e}(K^{(2)}, C_x^{(1)})\cdot\hat{e}(K_{\rho(x)}^{(4)}, C_x^{(2)})\right)^{\mu_x} \qquad (3)\\
= \;& \prod_{x\in I}\left(\hat{e}(g^r R_3', g^{a\lambda_x}T_{\rho(x)}^{-r_x})\cdot\hat{e}(T_{\rho(x)}^r R_{3,\rho(x)}, g^{r_x})\right)^{\mu_x}\\
= \;& \prod_{x\in I}\left(\hat{e}(g,g)^{a\lambda_x r}\cdot\hat{e}(g, T_{\rho(x)})^{-rr_x}\cdot\hat{e}(g, T_{\rho(x)})^{rr_x}\right)^{\mu_x}\\
= \;& \hat{e}(g,g)^{ar\sum_{x\in I}\lambda_x\mu_x}\\
= \;& \hat{e}(g,g)^{ars}\\
= \;& A_3.
\end{aligned}$$

Then, from Equations (1), (2) and (3), we will get $K = \hat{e}(g,g)^{\alpha s} = A_1/A_2A_3$ so as to recover the real messages. The correctness can be checked easily.

Delegate. The key delegation algorithm takes as input an attribute set ω' private key $K^{(1)}$, $K^{(2)}$, $\{K_j^{(3)}\}_{j\in\mathcal{U}\setminus\{k\}}$ and $\{K_i^{(4)}\}_{i\in\omega}$ for attribute set ω and index $k \in \mathcal{U}$ where $\omega' \subset \omega$. It chooses a random $r' \in \mathbb{Z}_N$ and $\tilde{R}_3, \tilde{R}_3', \tilde{R}_{3,i}, \tilde{R}_{3,j} \in G_{p_3}$, then computes the new key as:

$$\begin{aligned}
\tilde{K}^{(1)} &= K^{(1)}(g^a)^{r'}(u_k)^{r'}\tilde{R}_3, \quad \tilde{K}^{(2)} = K^{(2)}g^{r'}\tilde{R}_3',\\
\{\tilde{K}_j^{(3)} &= K_j^{(3)}u_j^{r'}\tilde{R}_{3,j}\}_{j\in\mathcal{U}\setminus\{k\}},\\
\{\tilde{K}_i^{(4)} &= K_i^{(4)}T_i^{r'}\tilde{R}_{3,i}\}_{i\in\omega'}.
\end{aligned}$$

Obviously, $\tilde{K}^{(1)}$, $\tilde{K}^{(2)}$, $\{\tilde{K}_j^{(3)}\}_{j\in\mathcal{U}\setminus\{k\}}$ and $\{\tilde{K}_i^{(4)}\}_{i\in\omega'}$ is a proper distributed re-randomized private key for index k and attribute set ω under random parameter $r + r' \in \mathbb{Z}_N$.

4 Security Analysis for CP-ABBE Scheme

To prove our CP-ABBE scheme is semantic secure in full security model, we employ the DSEM. Before conducting our proof, we present the semi-functional form of secret key and ciphertext (header).

Semi-Functional Key. There are two types of semi-functional key in our proof. Firstly, we get normal secret key for index t and attribute set ω as: $K'^{(1)}$, $K'^{(2)}$, $\{K_j'^{(3)}\}_{j\in\mathcal{U}\setminus\{t\}}$ and $\{K_i'^{(4)}\}_{i\in\omega}$. Then we form these two types of semi-functional key as follows: for each attribute $i \in \mathcal{N}$, we randomly pick a unique $z_i \in \mathbb{Z}_N$. These values will be used in both semi-functional key and ciphertext.

Type 1. We randomly choose $d, \sigma, \delta_j \in \mathbb{Z}_N$ for $i \in \mathcal{U}$ and $j \in \omega$, then compute:

$$\begin{aligned}
K^{(1)} &= K'^{(1)}g_2^d, K^{(2)} = K'^{(2)}g_2^\sigma,\\
K_j^{(3)} &= K_j'^{(3)}g_2^{\sigma\delta_j}, K_i^{(4)} = K_i'^{(4)}g_2^{\sigma z_i}.
\end{aligned}$$

Type 2. The Type 2 semi-functional key is generated as:

$$\begin{aligned}
K^{(1)} &= K'^{(1)}g_2^d, K^{(2)} = K'^{(2)},\\
K_j^{(3)} &= K_j'^{(3)}, K_i^{(4)} = K_i'^{(4)}.
\end{aligned}$$

Semi-Functional Ciphertext. A semi-functional ciphertext associated with broadcast index set $S = \{1, 2, \cdots, r\}$ and access structure $\mathbb{A}(A,\rho)$ is formed as follows: we first generate normal ciphertext $K', \{C_x'^{(1)}, C_x'^{(2)}\}_{\forall x}, C'^{(3)}, C'^{(4)}$. Then we randomly choose $\gamma, \gamma_x \in \mathbb{Z}_N$, a vector $\vec{u} = (u_1, u_2, \cdots, u_n) \in \mathbb{Z}_N^n$ and $\delta_c \in \mathbb{Z}_N$. Then we compute

$$\begin{aligned}
K &= K', \quad C_x^{(1)} = C_x'^{(1)}g_2^{A_x\vec{u}+\gamma_x z_{\rho(x)}},\\
C_x^{(2)} &= C_x'^{(2)}g_2^{-\gamma_x}, C^{(3)} = C'^{(3)}g_2^{\gamma\delta_c},\\
C^{(4)} &= C_j'^{(4)}g_2^{\gamma}.
\end{aligned}$$

Here, the value $z_{\rho(x)}$ is the correspondent value used in semi-functional key $\rho(x) = i \in \mathcal{N}$.

We can see that if semi-functional ciphertext is paired with semi-functional key, there exists a blind factor to prevent regular decryption operation.

We will prove the security of our scheme by sequence of games and hybrid argument. Firstly, we define these games between adversary and challenger. The first game $Game_{Real}$ is the same as the real security game. That is all the key and ciphertext are replied in normal form. In the next game, $Game_0$, all the replied keys are in normal form and the challenge ciphertext changes to semi-functional form. Then, without lose of generality, we let q denote the numbers of key generation query the adversary makes. We define:

- $Game_{k,1}$. In this game, the challenge ciphertext is semi-functional, the fist $k-1$ keys are Type 2 semi-functional key and the k'th key is semi-functional of Type 1 and rest of keys are replied in normal form.

- $Game_{k,2}$. In this game, the challenge ciphertext is semi-functional, the first k keys are in semi-functional of Type 2 and the following keys are in normal form. We note that in $Game_{q,2}$, all the keys are replied in Type 2 semi-functional form.

At last, we define the $Game_{Final}$ which is the same as $Game_{q,2}$ except the challenge ciphertext is semi-functional for a random message, other than neither of the two chosen messages by adversary, thus the adversary has no advantage. The advantage of adversary $\mathcal{A}$ in $Game_*$ is denoted by $Game_*Adv_{\mathcal{A}}$. Note, here we have $Game_{Real}Adv_{\mathcal{A}} = Adv_{\mathcal{A}}^{CP-ABBE}(\lambda)$ for some fixed security parameter λ.

Lemma 1. *Suppose there exists an algorithm $\mathcal{A}$ such that $Game_{Real}Adv_{\mathcal{A}} - Game_0Adv_{\mathcal{A}} = \epsilon$. Then we can build an Algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 1.*

Proof. We show how to construct an algorithm $\mathcal{B}$ to break Assumption 1 by interacting with adversary $\mathcal{A}$. $\mathcal{B}$ first receives the challenge (g, X_3, T). It must decide $T \leftarrow_R G_{p_1}$ or $T \leftarrow_R G_{p_1p_2}$. $\mathcal{B}$ chooses random exponents $a, \alpha, a_j, t_i \in \mathbb{Z}_N$ for $j \in \mathcal{U}$ and $i \in \mathcal{N}$. Then it forms the public parameters PK as:

$$\left(N, g, \hat{e}(g,g)^{\alpha}, g^a, \{u_j = g^{a_j}\}_{j\in\mathcal{U}}, \{T_i = g^{t_i}\}_{i\in\mathcal{N}}\right).$$

When $\mathcal{A}$ requires the secret key for any attribute set ω and index $t \in \mathcal{U}$, $\mathcal{B}$ can form it in normal form readily since it knows the master key α and X_3.

$\mathcal{A}$ submits an access structure $\mathbb{A}(A^*, \rho)$ and a broadcast index set $S^* \subset \mathcal{U}$ to challenger $\mathcal{B}$. $\mathcal{B}$ flips a coin $b \in \{0,1\}$, randomly picks $r'_x \in \mathbb{Z}_N$ for all A^*_x and a random vector $\vec{v'} = (1, v'_2, \cdots, v'_n) \in \mathbb{Z}_N^n$ and computes:

$$\begin{aligned} K_0 &= \hat{e}(T,g)^{\alpha}, \quad C_x^{(1)} = T^{aA^*_x\vec{v'}}T^{-r'_x t_{\rho(x)}}, \\ C_x^{(2)} &= T^{r'_x}, \quad C^{(3)} = T^{\sum_{j\in S^*} a_j}, \quad C^{(4)} = T. \end{aligned}$$

Then it chooses $K_1 \leftarrow_R \mathcal{K}$ and sends $\mathcal{A}$ $\langle Hdr^*, K_b\rangle$, where $Hdr^* = \left(\{C_x^{(1)}, C_x^{(2)}\}_{\forall x}, C^{(3)}, C^{(4)}\right)$.

If $T \in G_{p_1p_2}$, $\mathcal{B}$ implicitly sets g^s and g_2^{γ} is the G_{p_1} and G_{p_2} part of T respectively. This sets $\vec{v} = s\vec{v'}$ and thus $A^*_x\vec{v}$ is a proper share of secret s. Meanwhile, we can see these results is a properly distributed semi-functional ciphertext with $\vec{u} = a\gamma\vec{v'}$, $r_x = sr'_x$, $\gamma_x = \gamma r'_x$, $z_{\rho(x)} = t_{\rho(x)}$ and $\delta_c = \sum_{j\in S^*} a_j$. These values modulo p_2 are uncorrelated from their values modulo p_1 due to Chinese Remainder Theorem, thus we are able to reuse the values in G_{p_2} part.

Otherwise, if $T \in G_{p_1}$, this is a normal ciphertext under random encryption exponent s. Therefore, if $\mathcal{A}$ can distinguish these two games then $\mathcal{B}$ will distinguish the two distribution so as to break the assumption. □

Lemma 2. *Suppose there exists an algorithm $\mathcal{A}$ such that $Game_{k-1,2}Adv_{\mathcal{A}} - Game_{k,1}Adv_{\mathcal{A}} = \epsilon$. Then we can build an Algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.*

Proof. Now we also show how to construct a simulation algorithm $\mathcal{B}$, using the advantage of adversary $\mathcal{A}$, to break Assumption 2. $\mathcal{B}$ first receives the challenge $(G, g, X_1X_2, X_3, Y_2Y_3, T)$ and it should decide $T \leftarrow_R G$ or $T \leftarrow_R G_{p_1p_3}$. $\mathcal{B}$ chooses random exponents $a, \alpha, a_j, t_i, b \in \mathbb{Z}_N$ for $j \in \mathcal{U}$ and $i \in \mathcal{N}$. Then it forms the public parameters PK as:

$$\left(N, g, \hat{e}(g,g)^{\alpha}, g^a, \{u_j = g^{a_j}\}_{j\in\mathcal{U}}, \{T_i = g^{t_i}\}_{i\in\mathcal{N}}\right).$$

For the i'th key generation query on index t and attribute set ω, where $i < k$, made by $\mathcal{A}$, $\mathcal{B}$ randomly chooses $h, r \in \mathbb{Z}_N$, generates $R'_3, \{R_{3,j}\}_{j\in\mathcal{U}\setminus\{t\}}, \{R_{3,i}\}_{i\in\omega} \in G_{p_3}$ by using X_3 and sets the semi-functional key as:

$$\begin{aligned} K^{(1)} &= g^{\alpha+ar}u_t^r(Y_2Y_3)^h, \quad K^{(2)} = g^r R'_3, \\ K_j^{(3)} &= u_j^r R_{3,j}, \quad K_i^{(4)} = T_i^r R_{3,i}. \end{aligned}$$

It is easy to see that this is a well-distributed Type 2 semi-functional key where $\mathcal{B}$ implicitly sets $Y_2^h = g_2^d$. For $i > k$, $\mathcal{B}$ can simply runs the key generation algorithm since it knows the master key.

For k'th key query, $\mathcal{B}$ generates $R_3, R'_3, \{R_{3,j}\}_{j\in\mathcal{U}\setminus\{t\}}, \{R_{3,i}\}_{i\in\omega} \in G_{p_3}$ and "programs" the challenge parameter into this replied key as:

$$K^{(1)} = g^{\alpha}T^{a+a_t}R_3, \quad K^{(2)} = TR'_3,$$

$$K_j^{(3)} = T^{a_j}R_{3,j}, \quad K_i^{(4)} = T^{t_i}R_{3,i}.$$

Here, $\mathcal{B}$ implicitly sets g^r is the G_{p_1} part of T. If $T \in G$, it sets g^{σ} is the G_{p_2} part of T. Then we can see this is a proper Type 1 semi-functional key where $d = \sigma(a + a_t), z_i = t_i$ and $\delta_j = a_j$ modulo p_2. Here, t_i and a_i modulo p_2 is uncorrelated from the value of t_i and a_i modulo p_1. Otherwise, if $T \in G_{p_1p_3}$, this is a normal key.

Consequently, $\mathcal{A}$ submits an access structure $\mathbb{A}(A^*, \rho)$ and a broadcast index set $S^* \subset \mathcal{U}$. $\mathcal{B}$ flips a coin $b \in \{0,1\}$ and sets $X_1X_2 = g^s g_2^{\gamma}$ implicitly, chooses random values $r'_x \in \mathbb{Z}_N$ for all A^*_x. Then it randomly chooses $u_2, u_3, \cdots, u_n \in \mathbb{Z}_N$ to form a random vector $\vec{u'} = (a, u_2, \cdots, u_n) \in \mathbb{Z}_N^n$ and prepares challenge ciphertext as:

$$\begin{aligned} K_0 &= \hat{e}(X_1X_2, g)^{\alpha}, \\ \{C_x^{(1)} &= (X_1X_2)^{A^*_x\vec{u'}}(X_1X_2)^{-r'_x t_{\rho(x)}}, \\ C_x^{(2)} &= (X_1X_2)^{r'_x}\}_{\forall x}, \\ C^{(3)} &= (X_1X_2)^{\sum_{j\in S^*} a_j}, C^{(4)} = X_1X_2. \end{aligned}$$

Then it chooses $K_1 \leftarrow_R \mathcal{K}$ and sends $\mathcal{A}$ $\langle Hdr^*, K_b, \rangle$ where $Hdr^* = \left(\{C_x^{(1)}, C_x^{(2)}\}_{\forall x}, C^{(3)}, C^{(4)}\right)$.

This sets vector $\vec{v}$ as $sa^{-1}\vec{u'}$ and $\vec{u} = \gamma\vec{u'}$. So s is being shared in G_{p_1} and γa is being shared in G_{p_2}. Meanwhile, this also sets $r_x = sr'_x$, $\gamma_x = -\gamma r'_x$, $z_{\rho(x)} = t_{\rho(x)}$ and $\delta_c = \sum_{j\in S^*} a_j$ modulo p_2. Thus, above challenge ciphertext is almost a properly distributed semi-functional ciphertext and we should state following two problems.

At this point, challenger $\mathcal{B}$ seems able to construct a semi-functional ciphertext to test whether the k'th key is in normal form or in semi-functional form. However, in our reduction, this problem is not aroused. Firstly, note that we force $\mathcal{B}$ only to be able to generate semi-functional ciphertext with the same type as above challenge ciphertext. If we have $\omega \in \mathbb{A}(A^*, \rho)$ and $t \in S^*$ for this ciphertext and k'th key is in semi-functional form, then in decryption procedure, decryption Equation (2) will leave an additional term $\hat{e}(g_2, g_2)^{\sigma\gamma\delta_c}$ and Equation (3) will leave an additional term $\hat{e}(g_2, g_2)^{\sigma\gamma a}$. These two factors will cancel the G_{p_2} part of Equation (1), which is $\hat{e}(g_2, g_2)^{(d+\sigma\sum_{j\in S^*\setminus\{t\}}\delta_j)\gamma}$ (Note, the values of d, δ_c, δ_j are set properly). Meanwhile, if k'th key is in normal form, it can decrypt such testing ciphertext regularly. Therefore, the decryption test will successed unconditionally wether k'th key is in normal form or in semi-functional form.

Another problem is that value a appears both in k'th key's G_{p_2} part (i.e. $g_2^{\sigma(a+a_t)}$) and challenge ciphertext's G_{p_2} part (i.e. $g_2^{\gamma(A_x^*\vec{u'}-r'_x z_{\rho(x)})}$). We must argue that if adversary $\mathcal{A}$ dose not require the k'th key for index t which can decrypt the challenge ciphertext, then this correlation is also undetectable for it. There are two situations:

1) $\omega \notin \mathbb{A}(A^*, \rho) \wedge t \notin S^*$ or $\omega \in \mathbb{A}(A^*, \rho) \wedge t \notin S^*$ In this case, $\mathcal{A}$ cannot cancel $g_2^{a_t}$ out, thus $g_2^{\sigma(a+a_t)}$ is uniformly distributed under $\mathcal{A}$'s view since a_t are uniformly distributed and unknown to $\mathcal{A}$.

2) $\omega \notin \mathbb{A}(A^*, \rho) \wedge t \in S^*$ In this case, by using the decryption Equations (1) and (2), $\mathcal{A}$ may get $\hat{e}(g_2, g_2)^{sra}$. We argue $\mathcal{A}$ can not relate shares $g_2^{\gamma(A_x^*\vec{u'}-r'_x z_{\rho(x)})}$ with a in this part. Since $\omega \notin \mathbb{A}(A^*, \rho)$, vector $(1, 0, \cdots, 0) \in \mathbb{Z}_N^n$ is not in the span of rowspace $\boldsymbol{R}$ formed by row vector of A_x^* where $\rho(x) \in \omega$. Then there must be a vector $\vec{\omega}$ which is orthogonal to $\boldsymbol{R}$ and not orthogonal to vector $(1, 0, \cdots, 0)$. We find a basis $\boldsymbol{K}$ including $\vec{\omega}$ and set $\vec{u'} = f\vec{\omega} + \vec{u''}$ for some $f \in \mathbb{Z}_N$ and $\vec{u''}$ uniformly distributed in the span of other basis elements in $\boldsymbol{K}$. Then from row A_x^* that $\rho(x) \in \omega$, $\mathcal{A}$ can only get $A_x^*\vec{u''}$ from $A_x^*\vec{u'}$ and hence no information about f. Another potential way to get information about f is from equations $g_2^{\gamma(A_x^*\vec{u'}-r'_x z_{\rho(x)})}$. However, r'_x is not congruent to 0 modulo p_2 with overwhelming probability and $t_{\rho(x)} = z_{\rho(x)}$ for each x without appearing elsewhere (recall, we set unique z_i for each attribute $i \in \mathcal{N}$). Thus no information is got by $\mathcal{A}$ whose attributes of key satisfy $\omega \notin \mathbb{A}(A^*, \rho)$. in this part.

Above all, if $T \in G_{p_1p_3}$, then $\mathcal{B}$ has properly simulated $Game_{k-1}$. If $T \in G$, then $\mathcal{B}$ has properly simulated $Game_k$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

Lemma 3. *Suppose there exists an algorithm $\mathcal{A}$ such that $Game_{k,1}Adv_{\mathcal{A}} - Game_{k,2}Adv_{\mathcal{A}} = \epsilon$. Then we can build an algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.*

Proof. This proof is very similar to the proof of the previous lemma. After receiving the challenge parameters, $\mathcal{B}$ chooses random exponents $a, \alpha, a_j, t_i \in \mathbb{Z}_N$ for $j \in \mathcal{U}$ and $i \in \mathcal{N}$. Then it forms the public parameters PK as:

$$\left(N, g, \hat{e}(g,g)^{\alpha}, g^a, \{u_j = g^{a_j}\}_{j\in\mathcal{U}}, \{T_i = g^{t_i}\}_{i\in\mathcal{N}}\right).$$

$\mathcal{A}$ forms first $k-1$ secret keys and challenge ciphertext as previous Lemma and forms last $q-k$ keys by employing master key respectively. For k'th key on attribute set ω and user index t, $\mathcal{B}$ chooses random value $h \in \mathbb{Z}_N$, $R_3, R'_3, \{R_{3,j}\}_{j\in\mathcal{U}\setminus\{t\}}, \{R_{3,i}\}_{i\in\omega} \in G_{p_3}$ and sets:

$$\begin{aligned} K^{(1)} &= g^{\alpha}T^{a+a_t}R_3(Y_2Y_3)^h, \quad K^{(2)} = TR'_3, \\ K_j^{(3)} &= T^{a_j}R_{3,j}, \quad K_i^{(4)} = T^{t_i}R_{3,i}. \end{aligned}$$

This sets g^r is the G_{p_1} part of T. It is easy to see if $T \in G_{p_1p_3}$, this is a well-formed Type 2 semi-functional key and $\mathcal{B}$ has properly simulated $Game_{k,2}$. Otherwise, $T \in G$, this is Type 2 semi-functional key and $\mathcal{B}$ has properly simulated $Game_{k,1}$. In both case the decryption test will fail since the random term $(Y_2Y_3)^h$ cannot be cancelled out. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

Lemma 4. *Suppose there exists an algorithm $\mathcal{A}$ such that $Game_{q,2}Adv_{\mathcal{A}} - Game_{Final}Adv_{\mathcal{A}} = \epsilon$. Then we can build an algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.*

Proof. In this game, $\mathcal{B}$ first receives the challenge parameters $(N, g, g^{\alpha}X_2, X_3, g^sY_2, Z_2)$ and it needs to distinguish whether $T = \hat{e}(g,g)^{\alpha s}$ or $T \leftarrow_R G_T$. $\mathcal{B}$ chooses random exponents $a, a_j, t_i \in \mathbb{Z}_N$ for $j \in \mathcal{U}$ and $i \in \mathcal{N}$. Then it forms the public parameters PK as:

$$\left(N, g, \hat{e}(g^{\alpha}X_2, g), g^a, \{u_j = g^{a_j}\}_{j\in\mathcal{U}}, \{T_i = g^{t_i}\}_{i\in\mathcal{N}}\right).$$

Here we note that $\hat{e}(g^{\alpha}X_2, g) = \hat{e}(g,g)^{\alpha}$.

In key generation phase, for attribute set ω and user index t, $\mathcal{B}$ picks $r, t \in \mathbb{Z}_N$, $R_3, R'_3, \{R_{3,j}\}_{j\in\mathcal{U}\setminus\{t\}}, \{R_{3,i}\}_{i\in\omega} \in G_{p_3}$ randomly and sets Type 2 semi-functional key as:

$$\begin{aligned} K^{(1)} &= g^{\alpha+ar}u_t^r z_2^t R_3, \quad K^{(2)} = g^r R'_3, \\ K_j^{(3)} &= u_j^r R_{3,j}, \quad K_i^{(4)} = T_i^r R_{3,i}. \end{aligned}$$

$\mathcal{A}$ sends $\mathcal{B}$ an access structure $\mathbb{A}(A^*, \rho)$ and a broadcast index set $S^* \subset \mathcal{U}$. $\mathcal{B}$ flips a coin $b \in \{0, 1\}$ and chooses random values $r'_x \in \mathbb{Z}_N$ for all A_x^*. Then, it forms a random vector $\vec{u'} = (a, u_2, \cdots, u_n) \in \mathbb{Z}_N^n$ and sets challenge ciphertext as:

$$\begin{aligned} K_0 &= T, \quad C_x^{(1)} = (g^sY_2)^{A_x^*\vec{u'}}(g^sY_2)^{-r'_x t_{\rho(x)}}, \\ C_x^{(2)} &= (g^sY_2)^{r'_x}, C^{(3)} = (g^sY_2)^{\sum_{j\in S^*} a_j}, C^{(4)} = g^sY_2. \end{aligned}$$

This implicitly sets $g^sY_2 = (g^s g_2^{\gamma})$, $r_x = \gamma'_x\tilde{s}$, $\gamma_x = \gamma r'_x$, $\vec{v} = sa^{-1}\vec{u'}$ and $\vec{u} = \gamma\vec{u'}$. Thus, if $T = \hat{e}(g,g)^{\alpha s}$, this is a well distributed semi-functional ciphertext under random

Table 1: Security and function comparisons between ABBE schemes

Scheme	Access Structure	Full security	Standard Model
[1] (Scheme 3)	Monotone	No	Yes
[13]	AND&NOT	No	No
[9]	DNF&CNF	No	No
[19]	AND	No	Yes
This Work	Monotone	Yes	Yes

Table 2: Efficiency comparisons between ABBE schemes

Scheme	|PK|	|SK|	|CT|	Pairing
[1] (Scheme 3)	$\mathcal{O}(m+n)$	$\mathcal{O}(k)$	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$
[13]	$\mathcal{O}(n)$	$\mathcal{O}(k)$	$\mathcal{O}(n-r)$	$\mathcal{O}(1)$
[9]	$\mathcal{O}(m+n)$	$\mathcal{O}(m+n)$	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$
[19]	$\mathcal{O}(\log(n)+m)$	$\mathcal{O}(k)$	$\mathcal{O}(1)$	$\mathcal{O}(\ell)$
This Work	$\mathcal{O}(m+n)$	$\mathcal{O}(n+k)$	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$

encryption exponent s. Otherwise, it is a properly distributed semi-functional encryption of a random message in G_T. Thus, $\mathcal{B}$ can use the output $\mathcal{A}$ to gain advantage ϵ in breaking Assumption 3. □

Theorem 1. *If assumptions 1, 2, 3 hold, then our CP-ABBE scheme is fully secure.*

Proof. If Assumptions 1, 2, and 3 hold, then we have shown by the previous lemmas that $Game_{Real}$ is indistinguishable from $Game_{Final}$, in which the bit value of b is information theoretically hidden from the adversary. Therefore, the adversary can get no advantage in breaking our revocation scheme (i.e. $Adv_{\mathcal{A}}^{CP-ABBE}(\lambda)$ is negligible). □

5 Discussions

In this section, we first discuss the solution of the problem proposed in Section **1**. Then, comparison between the existent ABBE schemes is provided.

By recalling the example of hierarchical management structure the university, we show how to employ our construction to provide such appropriate solution. Firstly, university manager generates the key for each school. Assume the index of one school is k, then it may receive the key with form: $K^{(1)} = g^{\alpha+ar}u_k^r R_3$, $K^{(2)} = g^r R_3'$, $\{K_j^{(3)} = {u_j}^r R_{3,j}\}_{j\in\mathcal{U}\setminus\{k\}}$, $\{K_i^{(4)} = T_i^r R_{3,i}\}_{i\in\mathcal{N}}$. Note, the school k gets all the attribute key material $T_i^r R_{3,i}$. When some student in school k applies for attribute key, depending on her attributes, school k runs key delegation algorithm to generate the proper one and passes it to her. For the broadcast message to some intended students, the sender may specify the school index which they belong to and the access structure to generate message encryption key and broadcast header. Above solution is more efficient and flexible than traditional CP-ABE scheme where the attribute key distribution task is distributed to the second level manager thus avoid the heavy computation cost and storage overhead for one party.

In Table 1, we give the comparisons between our construction and mentioned ABBE schemes for security level and functionality. Item "Access Structure" denotes the policy which is supported by the scheme. "Full Security" denotes whether such scheme can be proved in fully secure model. "Standard Model" indicates whether the security of scheme can be based on static computational assumption without relying random oracle. Again, in Table 2, we give a rough efficiency comparison among these schemes since the goal and constructions of theirs is somewhat different. In this table, we employ k and m to express total number of users and size of universal attribute set respectively. k denotes the size of attribute set and ℓ denotes the size of access structure. r is the number of intended indexes. |PK|, |SK| and |CT| denote the overhead of public parameters, private key and ciphertext and here the ciphertext consists of symmetric encryption key and header. We measure the decryption efficiency by most costly pairing calculation which is denoted by "Pairing".

From the comparisons, it is apparent to see that our scheme enjoys high level security, expressive formula (any monotone access structure can realizes and thus implies AND, DNF and CNF formula) and high efficiency. Some scheme requires only constant ciphertext overhead or pairing calculations. However, they just realize restricted access policy.

6 Conclusions

In this paper, the first fully secure attribute based broadcast encryption (ABBE) scheme is proposed. Besides traditional BE schemes, it enables more flexible access policy to control the broadcast data. Its IND-CPA security can be reduced to three static intractable computational assumptions through recent dual system encryption proof technology in *fully secure model*. Some potential applica-

tions of our ABBE scheme are also presented. Comparing with the existent ABBE schemes, our construction achieves more flexible access control functionality, high security level and efficiency.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Pairing-Based Cryptography V Pairing '09*, pp. 248–256, Aug. 2009.

[2] E. A. Quaglia B. Libert, K. G. Paterson, "Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model," in *Public Key Cryptography*, LNCS 7293, pp. 206–224, Springer-Verlag, 2012.

[3] A. Beimel, "Secure schemes for secret sharing and key distribution," Israel Institute of Technology, Haifa, Israel, 1996.

[4] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proceedings of Advances in Cryptology (Crypto'05)*, pp. 258–275, Santa Barbara, California, USA, Aug. 2005.

[5] D. Boneh and E. Goh K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of Cryptography*, pp. 325–342, Cambridge, MA, USA, Feb. 2005.

[6] D. Cécile, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Advances in Cryptology (Asiacrypt'07)*, LNCS 4833, pp. 200–215, Springer-Verlag, 2007.

[7] A. Fiat and M. Naor, "Broadcast encryption," in *Proceedings of Advances in Cryptology (Crypto'93)*, pp. 480–491, Santa Barbara, California, USA, Aug. 1993.

[8] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems," in *Proceedings of Advances in Cryptology (Eurocrypto'09)*, pp. 171–188, Cologne, Germany, Apr. 2009.

[9] P. Junod and A. Karlov, "An efficient public-key attribute-based broadcast encryption scheme allowing arbitrary access policies," in *Proceedings of the 10th Annual ACM Workshop on Digital Rights Management*, pp. 13–24, Chicago, Illinois, USA, Oct. 2010.

[10] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proceedings of Advances in Cryptology (Crypto'10)*, pp. 62–91, Santa Barbara, CA, USA, Aug. 2010.

[11] A. Lewko, A. Sahai, and B. Waters, "Revocation systems with very small private keys," in *Proceedings of the 31th IEEE Symposium on Security and Privacy*, pp. 273–285, Berleley/Oakland, California, USA, May 2010.

[12] A. Lewko and B. Waters, "New techniques for dual system encryption and fully secure HIBE with short ciphertexts," in *The 7th Theory of Cryptography Conference*, pp. 455–479, Zurich, Switzerland, Feb. 2010.

[13] D. Lubicz and T. Sirvent, "Attribute-based broadcast encryption scheme made efficient," in *First International Conference on Cryptology in Africa*, pp. 325–342, Casablanca, Morocco, June 2008.

[14] B. Malek and A. Miri, "Adaptively secure broadcast encryption with short ciphertexts," *International Journal of Network Security*, vol. 14, no. 2, pp. 71–79, 2012.

[15] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM Conference on Computer and Communication Security*, pp. 195–203, Alexandria, VA, USA, Oct. 2007.

[16] M. Ramkumar, "Broadcast authentication with preferred verifiers," *International Journal of Network Security*, vol. 4, no. 2, pp. 166–178, 2007.

[17] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of Advances in Cryptology (Crypto'05)*, pp. 457–473, Santa Barbara, California, USA, Aug. 2005.

[18] X. Zhao and F. Zhang, "Analysis on hu et al.'s identity-based broadcast encryption," *International Journal of Network Security*, vol. 13, no. 3, pp. 178–180, 2011.

[19] Z. Zhou and D. Huang, "On efficient ciphertext-policy attribute based encryption and broadcast encryption," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pp. 753–755, New York, NY, USA, Oct. 2010.

Qinyi Li received his M.S. and B.S. degree from School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, P.R.China in 2013 and Southwest University of Science and Technology, Mianyang, P.R China in 2010 respectively. His research interests include: information security and cryptography.

Fengli Zhang is a professor in University of Electronic Science and Technology of China. She received her M.S. and Ph.D. degree from University of Electronic Science and Technology of China in 1986 and 2007 respectively. Her research interests include: information security and distributed mobile data management.

A Survey on Botnet Architectures, Detection and Defences

Muhammad Mahmoud[1], Manjinder Nir[2], and Ashraf Matrawy[2]
(Corresponding author: Muhammad Mahmoud)
Computer Engineering Department, King Fahd University of Petroleum & Minerals[1]
Dhahran 31261, Saudi Arabia
(Email: mimam@kfupm.edu.sa)
Department of Systems and Computer Engineering, Carleton University[2]
Ottawa, ON, K1S 5B6, Canada
(Received Sep. 7, 2012; revised and accepted Nov. 15, 2013)

Abstract

Botnets are known to be one of the most serious Internet security threats. In this survey, we review botnet architectures and their controlling mechanisms. Botnet infection behavior is explained. Then, known botnet models are outlined to study botnet design. Furthermore, Fast-Flux Service Networks (FFSN) are discussed in great details as they play an important role in facilitating botnet traffic. We classify botnets based on their architecture. Our classification criterion relies on the underlying C&C (Command and Control) protocol and thus botnets are classified as IRC (Internet Relay Chat), HTTP (HyperText Transfer Protocol), P2P (Peer-to-Peer), and POP3 (Post Office Protocol 3) botnets. In addition, newly emerging types of botnets are surveyed. This includes SMS & MMS mobile botnet and the botnets that abuse the online social networks. In term of detection methods, we categorize detection methods into three main groups, namely: (1) traffic behavior detection -in which we classify botnet traffic into; C&C traffic, bot generated traffic, and DNS traffic, (2) botmaster traceback detection, and (3) botnet detection using virtual machines. Finally, we summarize botnet defence measures that should be taken after detecting a botnet.

Keywords: Botnet, command and control, distributed denial of service attack (DDoS), fast-flux service networks

1 Introduction

A "botnet" is a term used to describe a network of infected hosts (**Bots**) which are running software robots and are being controlled by a human (**botherder**), via one or more controllers (**botmasters**). The botmaster's communication with its bots is called Command and Control (**C&C**) traffic [38, 46]. Botnets are a serious security threat. They are responsible for most of the email spam, identity theft, online phishing, online fraud, adware, spyware, and DDoS attacks [12]. It is estimated that about 15 percent of the computers connected to the Internet are infected and are used by botnets [3]. It has been documented that one botnet has infected and had more than 400,000 computers under its control [57]. Botnets have a very manipulative behavior, which makes their detection a challenging task. They can stay inactive for a very long time, and may generate a very low traffic volume [43, 57].

According to Bacher et al. [7], the attackers often target class B (/16) or smaller network ranges. Once these attackers compromise a machine, they install a botnet code (called *bot* in short) on it. The bot joins a specific communication server and listen to the C&C channel for further commands from its botmaster. This allows the attacker to remotely control this bot. Grizzard et al. [30] described the primary goals of botnets as follows: (1) **Information dispersion**; sending SPAM, Denial of Service (DoS) attack, providing false information from illegally controlled source. (2) **Information harvesting**; obtaining identity, financial data, password and relationship data. (3) **Information processing**; processing data to crack password for access to additional hosts.

Botnets are difficult to detect for many reasons; botnet's C&C traffic is usually low in volume, hidden in existing application traffic -which makes it look like normal traffic. The number of bots in a given network might be very low, or the botnet may use unusual destination port and/or encrypt its C&C traffic to avoid being detected [33, 45].

To demonstrate the potential of botnets Rajab et al. [56] provided multifaceted observations gathered from real world IRC botnets. To extract IRC specific feature, they used a binary analysis testing tool on gathered botnet traffic by using two independent means namely; IRC tracking and DNS cache probing, across the globe. They observed that to resolve IP addresses of their IRC server, most bots issue DNS queries. By tracking the botnet

traffic, they reported the following findings; (1) **Botnet Traffic Share**: The amount of botnet traffic is greater than 27% of all unwanted Internet traffic. They also observed that out of the 800,000 probed servers, 11% (85,000) were involved in at least one botnet activity. (2) **Botnet Spreading Behavior**: Like Worms, botnets continuously scan certain ports by following a specific target selection algorithm. After receiving a command over a C&C channel, botnets with variable scanning behavior start scanning for variabilities and this spreading behavior is more prevalent. (3) **Botnet Structure**: About 60% of all botnet traffic were IRC bots and only small percentage used HTTP for C&C. (4) **Effective Botnet Sizes**: Maximum size of online population is significantly smaller than the botnet footprint (number of hosts infected with the botnet). Moreover, they found that the botnet population depends on different time zones.

Botnets have been surveyed from different perspectives in the literature. Short overviews have been presented in workshops and conferences' surveys [8, 26, 40, 69]. Bailey et al. [8] focused on botnet detection and data sources. Their survey did not focus on botnet architecture or classification and no defence methods were surveyed. However, they surveyed botnet detection by cooperative behavior, attack behavior and signature based detection. Feily et al. [26] focused on botnet detection methods, but neither on botnets' architecture nor their classification, and no defence methods were surveyed. They [26] explained four botnet detection methods, namely; signature-based, anomaly-based, DNS-based and mining-based. Zhu et al. [69] put most of the focus on understanding botnet architecture anatomy where botnets were divided into IRC, HTTP, P2P botnets and fast-flux networks. Detection of botnets by honeynets and by traffic monitoring were mentioned. They [69] referenced a defence against spam and suggested a commercial security service for enterprises. In [40], Chao Li et al. presented a short survey on botnets and their evolution. They outlined botnets' infection mechanism, malicious behavior, C&C methods, communication protocols and they suggested some directions for botnet defence. In [63], Thing et al.'s survey focused on botnets that are used for DDoS attacks, and analyzed four DDoS attack botnets, and the way they launch their attacks. The article by J. Liu et al. [42] surveyed specific known botnets with their malicious activities and some detection methods. They highlighted IRC-based and P2P-based botnets. The majority of their survey focuses on some of the most popular botnet and their malicious activities. Their survey outlined four detection methods based on honeynets, IRC traffic analysis, IRC anomaly activities and DNS. They also referenced countermeasures for the public, home users and system administrators. Shin and Im [59] surveyed the threats and challenges of botnets. They classified botnet detection into C&C-based and P2P-based detection. They also outlined some defences against DDoS attacks. In a short survey [41], Lin and Peng focused on the detection methodologies and detection techniques of botnets. Among the various kinds of botnet attacks, they discussed three types of botnet attacks. Further, they described detection methodologies of botnet and surveyed two detection techniques. In [51], IBM published a report on botnets' risks and prevention. This report lists the security risks of botnets (such as DDoS, privacy, SPAM, phishing, etc.). Then, it focuses on IRC botnets and how they work. Finally, the report suggests some prevention measures against botnets and their risks. In 2011, the European Network and Information Security Agency (ENISA) published a report on botnet threats from industrial perspectives with focus on practical issues [53].

Our Contribution. Besides presenting an extensive survey on botnets and their detection mechanisms, we classified botnets based on underlying (C&C) infrastructure as well. Our survey has elaborated description of Fast-Flux service network and C&C rallying mechanism as both play very important roles in botnets' activities. To the best of our knowledge, this is the first survey that includes new emerging types of botnets like mobile and online social networks botnets. It also gives a comparison between current detection techniques as shown in Table 1 and explains the different attempts to create botnet behavior models.

Objective. The objective of this survey is to shed light on botnets threat by providing a clear background and classification on botnets architectures and their behavior, and to describe some security measures that are used to detect and mitigate botnets threats.

Survey Outline. This survey is outlined as follows: Section 2 lays some background on botnets architectures and mechanisms that botnets use to control other hosts. It analyzes botnets behavior and summarizes botnets models. Furthermore, it outlines some botnets facilitation features. Section 3 classifies botnets based on their underlying communication infrastructure, and surveys mobile botnets and online social networks botnets. In Section 4 we provide detail on botnets detection algorithms and classify them into; behavior-based, botmaster traceback-based, and virtual machine-based detections. We briefly illustrate some post-botnet detection measures. Finally, Section 5 concludes the survey.

2 Botnet Architectural Elements

In this section, we start by explaining what C&C is. Then, botnets infection behaviors and known botnets models are described. After that, we explained how fast-flux service networks work and surveyed botnets' C&C rallying mechanisms.

As Barford et al. [11] described, botnets usually have some of the following architecture features; They use existing protocols for their C&C communication (i.e. IRC, P2P, etc.). They may have the ability to exploit large

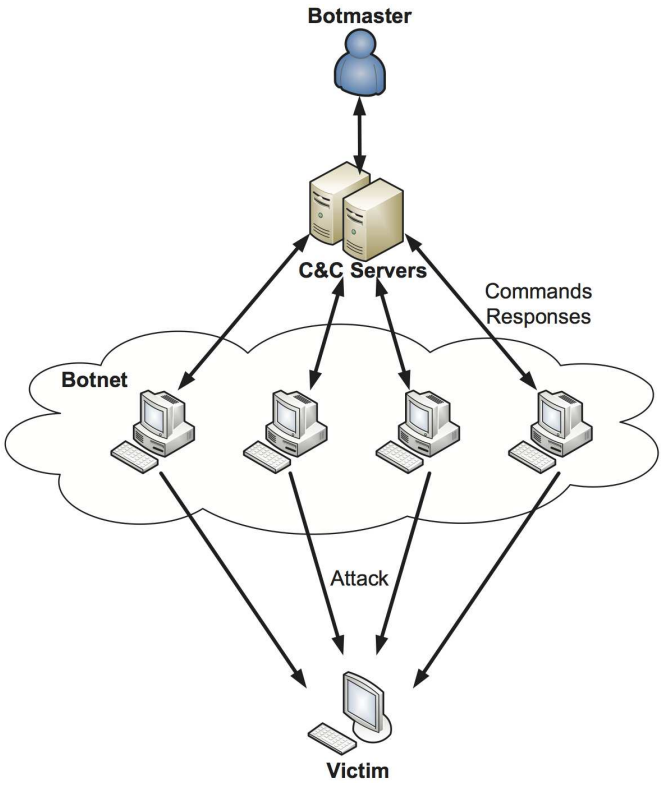


Figure 1: (IRC, HTTP, POP3) C&C architectures are usually centralized

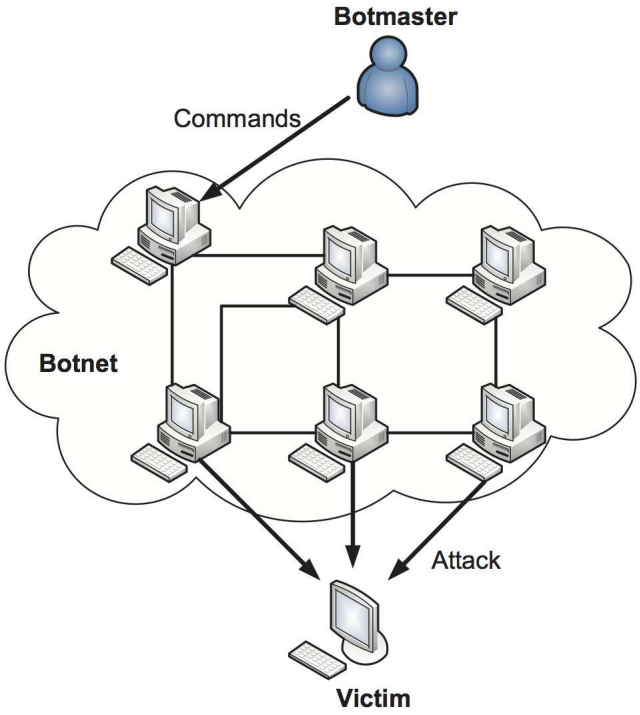


Figure 2: P2P C&C architectures are usually decentralized

number of targets, launch different types of DoS attacks, spy for passwords, fool the defence mechanisms, frustrate disassembly software, hide themselves from the local system, scan ports for vulnerabilities. Furthermore, botnets may encrypt their C&C traffic and/or may come with the C&C only and download other functionalities later on, as needed.

A victim host could be infected by targeting known vulnerability or by infected programs (like viruses). Once the host is infected, the bot can use any of the following mechanisms to control the infected host [11, 32, 52]; (1) secure the system (e.g. close NetBIOS shares, RPCDCOM), (2) spy or steal identity, (3) send SPAM emails, (4) host illegal sites, (5) redirect traffic for the botnet (e.g. fast-flux), (6) kill unwanted process running on the system (e.g. anti-virus, taskmanager, etc.), (7) test for virtual machines and/or debugger software, (8) add or delete autostart applications, (9) run or terminate programs, (10) download and execute files, (11) perform address and port scan, (12) rename files, (13) simulate key presses, (14) run DDoS attacks.

Furthermore, botnet propagation could be through horizontal or vertical scans. The horizontal propagation scan is done on a single port access for some address range. On the other hand, vertical scan is done on a single IP address across a range of ports [11, 32].

2.1 Command and Control

A botmaster's communication with the botnet is carried out via C&C. The C&C is the main feature that distinguishes botnets from other malwares [29]. It allows the botmaster to communicate with the botnet and give commands. Theoretically, the botmaster can command the botnet to do any task including; performing DDoS attacks, spamming, spying, identity theft, etc. [31, 36, 62].

To avoid detection, botnet designers tend to use widely used protocols for their C&C. Most botnets use IRC commands for their C&C communication [23, 51]. However, some botnets use the HTTP, POP3 or P2P protocols for their C&C communication. Newly emerging types of botnets use SMS, MMS, or online social networks for C&C communication.

The IRC, HTTP and POP3 botnets are usually centralized in the sense that their C&C channels depend on specific servers and if they are disabled, botnet will cease to exist. Figure 1 shows an architecture of botnets with centralized C&C server(s). On the other hand, P2P botnets are usually decentralized, as shown in Figure 2. In Section 3, we discuss botnets C&C in more detail as it is the main criterion of our botnets architecture classification.

2.2 Botnet Infection Behavior

Most botnets run in four phases [69]. A node's transition from a clean host to a zombie host, and reacting to its botmaster's commands, goes through four steps.

First, the initial infection starts when botnet nodes scan the network looking for vulnerabilities. They scan for back doors [69], known buffer overflows, known vulnerable network administrator tools. They may run brute force password scanning for some services (e.g. SQL servers, NetBIOS shares, etc.) [11]. Second, the secondary injection starts when a vulnerability is exploited and the victim host downloads and runs the bot's binary code [32]. Then, the bot establishes a connection to the botnet's C&C server, and starts to control the host (e.g. disable anti-virus, change NetBIOS shares, etc.). Finally, the malicious activities begin when the bot starts to act on botmaster's commands (e.g. run DoS attack, send SPAM, etc.) and then the botnet maintains and upgrades itself periodically [11].

In the case of P2P botnet, the first two steps are similar to other botnets. After the initial infection and injection, the P2P botnet uses an initial peer list to contact the initial peers. Once it finds a live peer, phase one starts where botnet updates its peer list and download any available updates. After that the node goes to phase two when it starts its malicious activities [27]. The aforementioned P2P botnet behavior is based on the STORM malware behavior. Other P2P malwares should -to some extend- have similar behavior [61].

2.3 Modeling Botnet Architectures

There has been different attempts at creating models for botnet behavior either to help understand botnets or to give the researchers a head start of possible future botnet designs. The following are examples of these models.

- **Diurnal Propagation Model**: A model by D. Dagon et al. [25] shows propagation dynamics in botnets and describes that time zones and geographical locations play a critical role in malware propagation. All the botnets studies use DNS to locate their C&C servers. However, through binary analysis, this model has confirmed that most botnets do not use hard coded IP addresses. In this model, an approach is used to predict botnet dynamics prior to an attack, and focuses on capturing any network cloud of coordinated attackers rather than tracking individual bots.
- **Super-Botnet Model**: Vogt et al. [67] stated that traditional botnets are easily detectable through their C&C. Therefore, they proposed a possibility of super-botnet, which is a network of independent botnets that can be co-ordinated for large scale attacks. To establish a super-botnet a two phase process is explained. The authors suggest that a super-botnet protects itself from defenders by not allowing individual botnets to have complete information about the super-botnet rather each botnet can have partial routing information to contact a small finite set of its neighbours.
- **Stochastic Peer-to-Peer Model**: Van et al. [58] presented a botnet stochastic model for the creation of a P2P botnet. The model was constructed in the Möbius [4] software tool. In this model, authors examine the growth of botnet size based on different parameters and suggest the development of future anti-malware systems against P2P botnets.
- **Advance P2P hybrid Model**: Keeping in view weaknesses of P2P botnet architecture, Wang et al. [68] proposed a design of an advanced hybrid P2P botnet architecture. The architecture is harder to be shut down or monitored by defenders. In this model, each bot has its individual encryption design and robust connectivity to other bots. The botnet can disperse communication traffic to different service ports in a way that the botnet will not be exposed if one of its bots is captured. Furthermore, the authors alarm us of advanced botnet attack techniques that could be developed by botmasters in the near future and propose honeypot to defend against such advanced botnets.

2.4 Botnet Facilitators

Botnets usually use some techniques to alleviate their activities. In this section, some of these techniques are surveyed.

2.4.1 Fast-Flux Service Networks

DNS is an Internet service which translates names of sites into their numeric IP addresses. Usually DNS do not respond to DNS requests with unique *'A'*[1] record. For every host, DNS has a list of *A* records each with a given Time-to-Live (TTL) value (normally from 1 to 5 days). DNS returns these *A* records in round-robin way [35]. This implementation of DNS is called Round-Robin DNS (RRDNS). Furthermore, in Content Distribution Networks (CDN)[2], the DNS is implemented in a sophisticated way that it finds out the nearest edge router and returns it to the client. CDN uses a much lower TTL value than RRDNS to enable them to react quickly to link changes. On the other hand, a Fast-Flux Service Network (FFSN) is a distributed proxy network -built on compromised machines (flux-agents)- that direct incoming DNS requests to the botnet's desired address on the fly [35].

Nazario et al. [49] and the Honeynet Project [7] discussed fast flux networks that are used as botnet C&C networks. Botnets use fast-flux DNS techniques to host unauthorized or illegal content within a botnet. This is done to allow the botnet's domain name to have multiple IP addresses. In the meantime, involved DNS records are constantly changing every few minutes using a combina-

[1] *'A'* record is a mapping between host name and IP address

[2] Also called content delivery network. It is a system in which many copies of data is placed in different location in the network. The purpose is to maximize the bandwidth, so when a user node tries to access some data, it will be directed to the server closest to it, rather than allowing all users access data on centralized server.

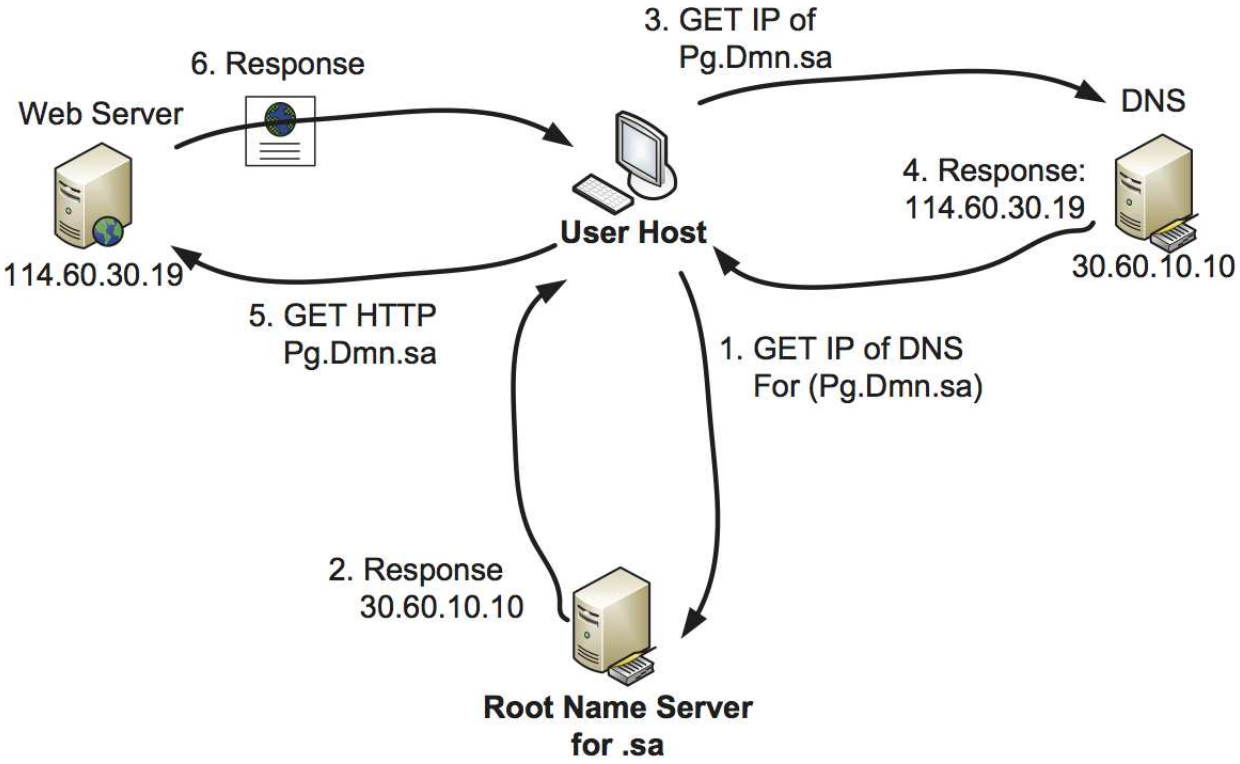


Figure 3: Normal content retrieval process

tion of round robin IP addresses and a very short TTL from any given particular DNS resource record (RR).

In FFSN, the victim client first sends an address query to DNS. Then, the DNS returns the IPs of a subset of active flux-agents. After that, the flux-agent relays the client's request to the mothership[3]. The key factor in FFSN is the combination of a very short TTL and the round-robin answer from a large pool of active agents. Because the TTL is short, the following DNS request will result in a totally different flux-agent. FFSN have high availability because the mothership continuously updates the pool of active agents. To have good understanding of fast-flux we need to learn all the steps for normal DNS query and ignoring steps that are unrelated to the fast-flux concept. The following steps describe the process of content retrieval -for web address "Pg.Dmn.sa"- in normal DNS queries. As Figure 3 shows, the web address is traversed from right to left. (1) The user host asks the ".sa" root name server for the IP address of the DNS responsible of the domain "Dmn.sa". (2) The ".sa" root name server replies with an IP address (30.60.10.10 in this case). (3) The user host then uses this IP address to contact to the DNS and ask it for the IP address of "Pg.Dmn.sa". (4) The DNS replies with an IP address (114.60.30.19 in this case). (5) The user host then uses this IP address to contact to the web server for the HTTP content of "Pg.Dmn.sa". (6) The web server responses with the requested contents [7, 35].

[3]a secret controlling element of the botnet.

As the botmaster tries to hide the IP address of unauthorized or illegal website(s), it tends to fast-flux their IP address(es). Figure 4 illustrates the steps of retrieving the content of fluxed web address "FlxPg.Dmn.sa". Steps (1) to (5) are identical to normal content retrieval steps, except that when the IP address of "FlxPg.Dmn.sa" is requested, the DNS response comes with a short TTL. Therefore, any subsequent DNS query would probably get a different IP address response. After the user host uses the IP address (114.60.30.19 in this case) to contact the "alleged webserver" requesting the contents of "FlxPg.Dmn.sa", this "alleged webserver" will carry out two more hidden steps. (5a) The "alleged webserver" will request the content of "FlxPg.Dmn.sa" from the mothership. (5b) The mothership responses with the requested contents. (6) The "alleged webserver' redirects the response from the mothership to the user host.

Sometimes, botmasters take one more step to make it more difficult to locate them by fluxing the IP address of the DNS too. Figure 5 illustrates the steps of retrieving the content of fluxed web address with fluxed DNS "FlxPg.FlxDmn.sa". (1) The user host asks the ".sa" root name server for the IP address of the DNS responsible of the domain "FlxDmn.sa". (2) The ".sa" root name server replies with an IP address (30.60.10.10 in this case) with short TTL. (3) The user host then uses this IP address to contact to the "alleged DNS" and ask it for the IP address of "FlxPg.FlxDmn.sa". (3a) The "alleged DNS" passes this request to the mothership. (3b) The mothership responses with an IP address (114.60.30.19 in this case) (4)

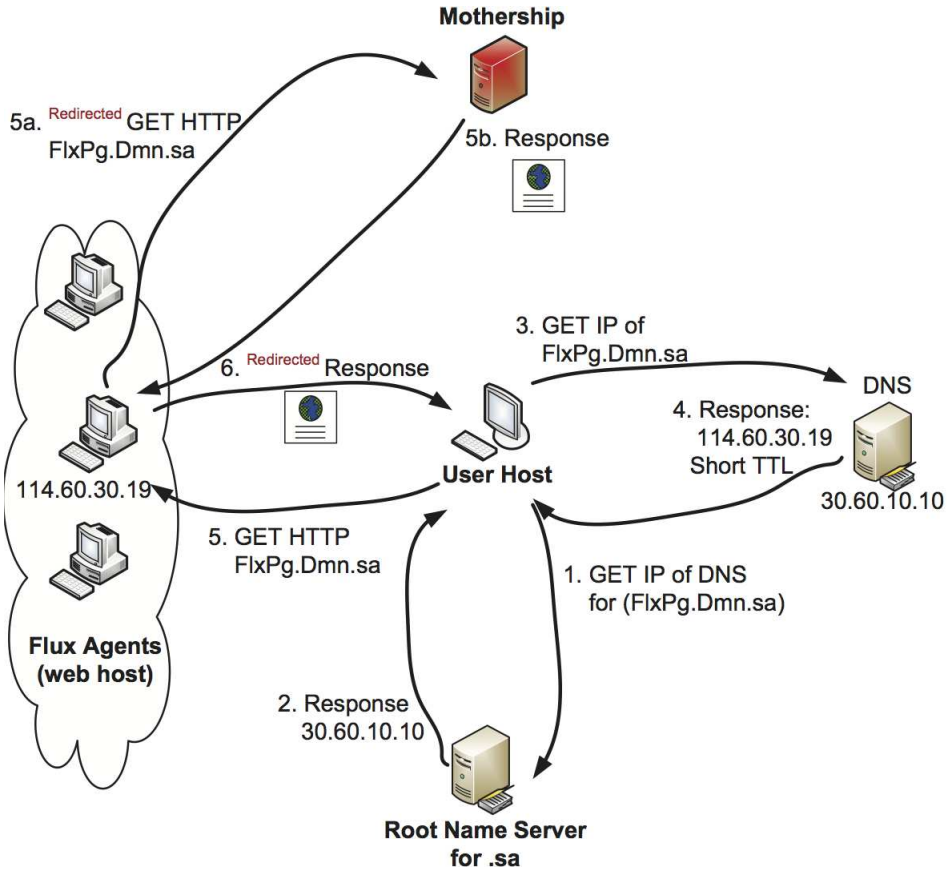


Figure 4: Single-Flux content retrieval process

The "alleged DNS" redirects the mothership's response to the user host. (5) The user host then uses this IP address to contact to an "alleged webserver" for the HTTP content of "FlxPg.FlxDmn.sa". (5a) The "alleged webserver" will request the content of "FlxPg.FlxDmn.sa" from the mothership. (5b) The mothership responses with the requested contents. (6) The "alleged webserver' redirects the response from the mothership to the user host.

To summarize, there are three types of fast-flux. (1) Single-Flux: when IP address of an unauthorized or illegal webpage is fluxed. (2) Name Server (NS)-Flux: when IP address of DNS is fluxed. (3) Double-Flux: when both IP addresses of the webpage and the DNS are fluxed [16]. Figures 3, 4, 5 illustrate the difference in content retrieval process between normal, Single-Flux and Double-Flux Service Networks.

Holz et al. [35] lists some features of FFSN that might help in detecting them. First, legitimate domains return 1 to 3 *A* records, but FFSN return 5 or more *A* records. Second, legitimate domains return a small number of nameserver (NS), but FFSN returns several NS records and several *A* records for the NS records. Third, legitimate domains return a small *A* records only from one autonomous system (AS), but FFSN tends to be located in more ASs. Furthermore, FFSN does not have the freedom to choose hardware and IP address. Therefore, the range of their IP addresses is diverse. Finally, since there is no physical agent control, therefore, there is no guaranteed up time [35, 52].

2.4.2 Command and Control Rallying Mechanisms

According to Choi et al. [20], botmasters want their bots to be invisible but portable, therefore they use different methods for bots rallying. They stated that not all bots can have mobility and invisibility at the same time. They described three rallying methods, namely; hard-coded IP address, dynamic DNS, and distributed DNS.

In hard-coded IP address method; the bot binary has a hard-coded IP address of its C&C server, the server can be detected through reverse engineering, and the botmaster can be quarantined or the botnet can be suspended. As hard-coded IP address cannot be changed, this method cannot provide mobility and does not make the botnet invisible as well. On the other hand, in dynamic DNS botnets migrate their C&C server frequently, upon the instruction of botmaster. Using a list of servers provided in the bot binary, a botmaster uses several C&C servers. It uses dynamic DNS in order not to be detected or suspended, and to keep the botnet portable. When connection to the C&C server fails or shutdown, the bots will perform DNS queries and will be redirected to a new C&C server [2]. This redirection behavior of botnets is known

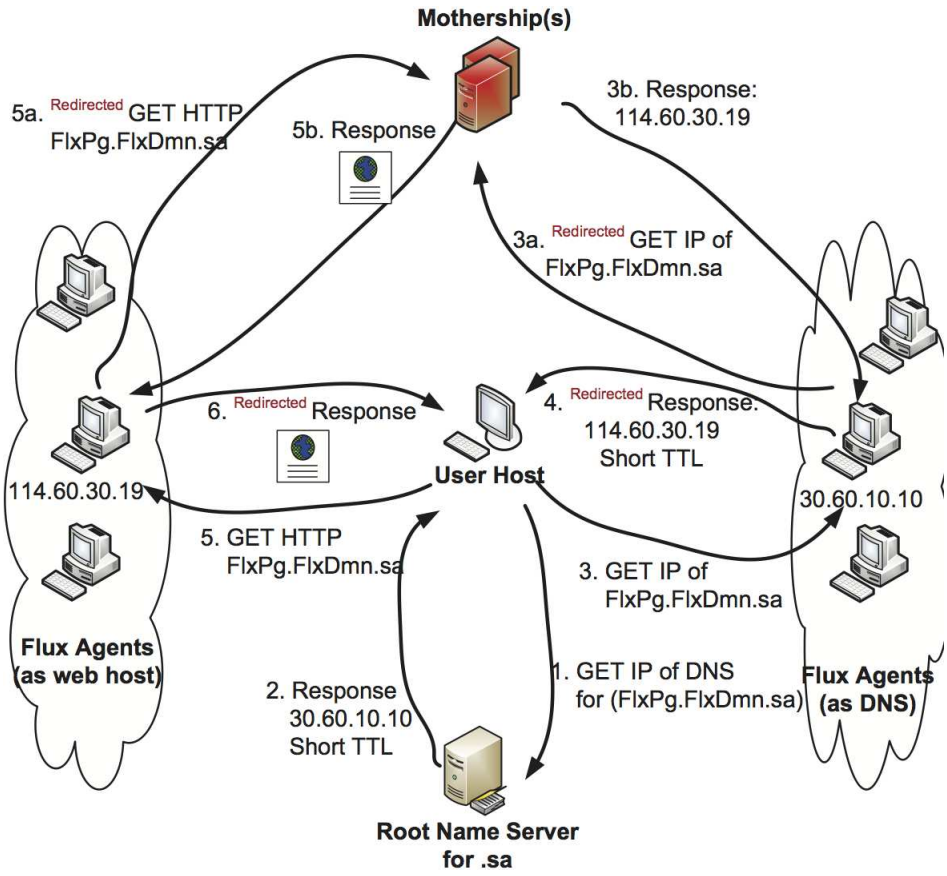


Figure 5: Double-Flux content retrieval process

as "herding". This method provides mobility and some invisibility to the botnets. Finally, with distributed DNS, botnets run their own distributed DNS service at locations that are out of the reach of law enforcement. Bots include the addresses of these DNS servers and contact these servers to resolve the IP address of C&C servers [2]. This method provides both mobility and invisibility to their botnets.

In summary, while the hard-coded IP botnet makes very easy for the newly infected nodes to join the botnet, it also makes easy for law enforcement to track and shutdown the botnet. On the other hand, using DNS to migrate C&C servers make it harder for the newly infected nodes to join the botnet. Some infected nodes might never be able to join the botnet -in case they stay offline long enough for all the addresses in the initial communication list to be obsolete, however, it gives the botnet the flexibility to hide its C&C severs.

3 Botnet Architecture Classification Based on C&C

In this section, we give a classification of botnets. Botnets can be classified based on their C&C traffic protocol.

3.1 IRC Botnets

The IRC [50] protocol was designed to facilitate a chatting environment. Its simplicity and distributed structure enables the earliest and most common botnets to use it for their C&C communication [23, 45]. IRC has many properties, which make it attractive for an attacker, such as, its redundancy, scalability and versatility. Furthermore, due to its long term and wide spread use, there is a large base of knowledge and source code to develop IRC-based bots [30].

As described by Cooke et al. [23], IRC is a well-known public exchange point and enables virtually instant communication, which provides a common, simple, low latency, wide availability and anonymity command and control protocol for bot communication. An IRC network is composed of one or more IRC servers. According to the botnet design, each bot connects to a public IRC network or a hidden IRC server. The bot receives commands from the controller (botmaster) and can be instructed to attack. The simplicity and multicast delivery mechanism of the IRC protocol fascinate attackers to use this protocol to send instructions and commands to bots.

Wang et al. [68] and Gizzard et al. [30] observed that most easily detected botnets use IRC for their C&C communication. They pointed out the weaknesses of IRC botnet because of its centralized server architecture. More-

over, IRC traffic is usually un-encrypted and once the centralized IRC C&C channel is detected by the defender the whole botnet could be disabled by shutting down the centralized server.

3.2 HTTP Botnets

In this botnet architecture, HTTP servers are used to distribute bot commands. Botnet members poll HTTP server(s) from time to time to get new commands [36].

Chiang et al. [19] have described botnets using HTTP as C&C mechanism. According to them, an HTTP bot is setup to communicate with certain webserver(s) using an HTTP post, which contains unique identifiers for the botnet, and in return the webserver will send the HTTP commands that it has been setup with. Afterwards, the bot could download malware files, spam information, or even DDoS instructions. The connection in HTTP botnet cannot be initiated from botmaster(s) as it does with the IRC botnets because botmasters and the bots are not constantly present on the HTTP channel. Compared to IRC, Nazario in [47] mentions that botnet designer could have two benefits of using HTTP for C&C communication. First, HTTP C&C is harder to detect as it blends into majority of traffic. Second, existing firewall policies block IRC C&C botnets, but HTTP based botnets can pass firewall policies. On the other hand, once the bot's HTTP server is identified, it can be isolated and shutdown [36].

3.3 POP3 Botnets

Singh et al. [60] have developed a prototype bot that demonstrates the feasibility of email-based botnet C&C. Jennifer Chandler [17] studied a bot that uses POP3 protocol for C&C communications. In a POP3 C&C architecture, the bot connects to a predefined mail server to retrieve an email message, which contains commands as email attachments and can respond to commands through the same channel. Chandler also mentioned that this traffic will be less detectable than a connection to IRC server. Similarly, Singh et al. [60], demonstrated that botnet commands can remain hidden in spam due to its enormous volume. If email service providers deploy specialized detection of spam-based botnets, botmasters can alternatively communicate with bots via non-spam email that cannot be safely discarded.

3.4 Peer-to-Peer Botnets

Holz et al. [36] explains a new decentralized architecture of botnets that is based on P2P protocols. P2P botnets are relatively new generation of botnets that do not use a central server to send C&C commands to botnet members. P2P botnets usually use publish/subscribe systems to communicate [36]. Unlike IRC botnets, the attacker in P2P botnets cannot send commands directly to bots, instead, a set of commands (C) is defined in the P2P system, and all bots subscribe to this set. When an attacker (any bot) wishes to launch an attack, it publishes a command (c_i) on the P2P system. All bots subscribed to the set will be able to see the command.

Barford et al. [11] have anticipated that future botnet development will include the use of encrypted C&C communication. As stated by Grizzard et al. [30], in a P2P architecture, there is no centralized point for C&C and bots communicate with other peer bots instead of a central server. Nodes in a P2P network act as both servers and clients. Therefore, there is no centralized coordination point that can be incapacitated. The authors analyzed the case study of the "Trojan.Peacomm" bot and observed that the P2P protocol is essentially being used as a name resolution server to upgrade the bot.

Wang et al. [68] pointed out some weaknesses of known P2P bots and proposed a new P2P bot architecture. According to them, botnets such as *Sinit*, *Phatbot*, *Nugache* and *Slapper* have implemented different kinds of P2P control architectures. A *Sinit* bot host finds other *Sinit* bot hosts by using random probing. The extensive probing traffic will make it easy to detect the botnet. *Phatbot* uses Gnutella cache servers for its bootstrap process. This also makes the botnet easy to be shut down. *Nugache* relies on a seed list of C&C IP addresses during its bootstrap process. This makes it weak. *Slapper* does not have encryption and its command authentication enables others to easily hijack it. Keeping in view these weaknesses of P2P botnet architecture Wang et al. propose the design of advanced hybrid P2P botnet architecture, which is much harder to be shut down or monitored. Their hybrid P2P botnet architecture provides robust network connectivity, individualized encryption, controlled traffic dispersion and easy monitoring and recovery by its botmaster. Furthermore, if a bot is captured, the botnet exposure will be limited.

Figure 2 shows an architecture diagram for P2P botnets. Unlike IRC or HTTP botnets, any bot in a P2P botnet can publish a command. Therefore, if one is able to identify a botmaster and bring it down, the P2P botnet will still be functional because any bot can issue botnet commands (i.e. be a botmaster).

3.5 Other Botnet Emerging Types

3.5.1 SMS & MMS Mobile Botnets

In this section, we are going to discuss botnets that spread on mobile devices and could use SMS or MMS for C&C communication.

Challenges. Mobile devices/smartphones pose some challenges which make them unattractive to botnet developers [24, 28]. They usually use the cellular network for communication. Thus they are not continuously available on the Internet. Even when smartphones access the Internet, they are usually either behind firewalls or using

dynamic private IP addresses that are not available on the Internet. This makes difficult to mobile bots to be visible to the botmaster. Therefore, botmasters will not be able to send direct commands to their bots. Furthermore, cellular networks do not use DNS to find mobile nodes. This will make botmaster invisible to botnets to make use of DNS services like Fast-Flux (see Section 2.4.1).

Compared to PCs, smartphones have limited power. Therefore, any abnormal consumption could lead to investigation that might detect the bot. Furthermore, data traffic or messaging cost is very noticeable in cellular networks. Therefore, if abnormal network traffic is generated because of the botnet, it can be easily noticed. Frequent IP change and not having long lived public IPs makes it impractical to deploy P2P-based C&C infrastructure [24]. In mobile botnets, botmaster could use traditional C&C methods (like IRC or HTTP) to communicates with its bots. However, cellular network architecture may limit the bots connectivity [65].

Effect. To overcome the challenge that mobile devices couldn't be reached directly by their botmasters, botmasters could use web servers to post their commands and updates [28]. Then, bots could access these web servers -when the mobile device is available on the Internet- to pull these commands [28]. Some botmaster could rely on the fact that most mobile devices have access to wireless LANs. This enables the botmaster to launch attacks on both, the Internet and the cellular network [64]. Mobile devices could send SMS and MMS to connect to their C&C proxy servers. In such case, these proxy servers would have access to the Internet and would be able to understand and read these SMS and MMS [28].

On the other hand, once a bot has control of a mobile device, it will be able to exploit services in the mobile network. It could monitor incoming messages and delete them before they appear in the inbox. Furthermore, as business users are using their mobile devices to access their banking information, store their sensitive information, order credit reports, and much more, which makes bot spying more threatening.

History. Botnets on mobile phones were almost unheard of until 2009 [24, 28]. In July 2009, Symantec [6] reported a mobile phone botnet that uses a "good old-fashioned social engineering mixed with SMS spam" to propagate. A phone will be infected when the user downloads from the external URL provided in an SMS. The bot hides itself in a process that has a name similar to a legitimate application. To defend itself, the bot is capable of ending applications that could allow the user to manually terminate the bot process. It connects to its server, using HTTP, to update itself and send spy information to the botmaster.

SRI international published a technical report [54] on "iKee.B", an iPhone bot that was captured on 25 November 2009. This bot is capable of checking its C&C server every five minutes to pull additional instructions and run their scripts on the hacked iPhone. It could also run scans on WiFi or some IP address range to infect other vulnerable iPhones. This is a very simple botnet that took very little memory of the iPhone. It has all the functionality that is expected of PC botnets, yet it has a flexible code base, which makes it very dangerous.

During RSA Conference 2010 in San Francisco, Derek Brown and Daniel Tijerina [34] demonstrated how they were able to develop a weather application that provides mobile users with weather forecast. However, to download this application, user had to approve some permission. According to [34], within an hour, there was 126 downloads of this application, then 702 downloads after eight hours. In few days, the number of smartphones running this application was about 8000. To show the danger of botnets on mobile phones, they wrote a malicious version of this application that can send spam, get user's physical addresses and contact information. It could also steal user's files, email, passwords, and access Facebook and Twitter accounts.

Xiang et al. [24] proposed a stealthy, resilient, low cost, Android-based botnet. They called it Andbot. They suggested a new centralized C&C topology called URL Flux. In URL Flux, there are a fixed number of C&C servers that can be accessed by the bot. This list of servers is built using Username Generation Algorithm (UGA). Therefore, the bot connects to a hardcoded Web 2.0 server (one of many) then traverse through a list of users generated by the UGA. Once a life user is found, Andbot commands can be deployed. This provides resilience to the mobile botnet since there are more than one centralized C&C server. To reduce the cost, Andbot avoids using SMS or Bluetooth for C&C. Instead it relies only on IP communication. Andbot uses RSS and GZIP to reduce its traffic.

In [64], Traynor et al. demonstrated how -by attacking the Home Locator Register (HLR) - a relatively small cellular botnet could cause a nation-wide outage in cellular network services. By launching network services requests (like $insert\_call\_forwarding$), the mobile botnet could cause serious degradation in the service without raising the attention of the bot-infected device owner [64].

3.5.2 Social Networks Botnets

In this section, we describe two types of online social networks (OSNs) botnets. The first type is the botnets that use existing (OSNs) for C&C communication (OSNs C&C Botnets). The other type is the botnets that are actually comprised of OSNs accounts (Socialbots)

OSNs C&C Botnets. In 2009, the first actual botnet that used social networks for C&C was reported by Jose Nazario [48]. The botmaster used accounts on Twitter.com and Jaiku.com to command bots to download and run malicious activities. In [39], Kartaltepe et al. studied this bot and found that it works as follows: The bot sends HTTP GET request to the botmaster's RSS. The return RSS feed has encoded message, which is decoded

by the bot to get one or more URLs. These URLs direct to a site with longer URLs each pointing to a malicious zip file. After downloading the zip file, the bot unzips and runs the malicious program. The malicious program collects user information and send it to the botmaster.

In 2009, Trend Micro published a technical report explaining the largest Web 2.0 botnet (KOOBFACE) [10]. KOOBFACE bot is comprised of many components. These components could be as follows: (1) **KOOBFACE downloader**: finds out the victims' social network, connects to C&C channel, and download other components. (2) **Social network propagation components**: this is the KOOBFACE worm that sends out the infection SPAM. (3) **Web server component**: makes the victim a web server for the KOOBFACE botnet. (4) **Ads pusher and rogue antivirus (AV) installer**: installs fake antivirus on the victim's machine and opens an ads window. (5) **CAPTCHA**[4] **breaker**: gets the victim to challenge-response tests. (6) **Data stealer**: steals product IDs, profiles, credentials, etc., and sends them in encrypted form to the C&C server. (7) **Web search hijackers**: intercepts the victim's search queries and redirects them to suspicious sites that returns result with some agenda. (8) **Rogue DNS changer**: modifies the victim's DNS to a fake one. This fake DNS intercepts the victim's web requests, delivers malware, and/or prevents the victim from accessing antivirus websites. KOOBFACE infection starts with a SPAM asking the user to watch a video. By clicking on the video's URL, the user will be directed to YuoTube (not YouTube) and will be asked to download an executable to watch the video. This executable is the aforemention KOOBFACE components downloader. KOOBFACE has an update capability that make it difficult to shutdown [10].

In [46], Nagaraja et al. designed a botnet that uses images shared by OSNs users for C&C communication. Bots in this botnets can communicate if they are hosted on computers for people using an OSN. When the botmaster issues a command, it uses images and post them on Facebook. When users of infected computers log into Facebook and view these images, the bot code on their computers intercepts these images and extracts the required information from them. On the other hand, to send stolen information to the botmaster, the bot waits for the user to post an image. It intercepts the the image and injects the data into it [46].

Socialbots. Socialbot is defined as "*an automation software that controls an adversary-owned or hijacked account on a particular OSN, and has the ability to perform basic activities such as posting a message and sending a connection request*" [13]. It is predicted that about 10% of social network individuals will be robots by 2015 [37]. The main character of the socialbot networks (SbN) is that instead of financial means, they focus on having substantial relationships with human users. They can cause peer effect to promote for a product or social engagements. Therefore, they have the potential for a great future opportunity [37]. For instance, to determine influences, some services score user's activity and their effect in the network. SbN can hijack and change these scores. SbN can learn the social graph, analyze people posts, decide what to say and to whom. They do that by posting and following. The digital space gives the SbN a "near-perfect" world to apply artificial intelligence theories. In Twitter, rate of friendship between user can be greatly changed using SbN [37].

For example, the Realboy Project compilation [22] is about designing a Twitter botnet that imitates human users with three main goals; (1) to repost external users tweets, (2) to follow other users, (3) to get 25% follow-back rate. In addition, Boshmaf et al. [13] designed and analyzed a socialbot. Their Facebook botnet had one botmaster, 102 bots, and they ran it for eight weeks. During this period, the socialbot sent 8570 friendship requests. 3055 out of these requests were accepted. They recorded related data and all accessible profile information. Boshmaf et al. [14] concluded that online social networks (OSN) are vulnerable to large scale infiltration and that SN defence systems do not try to prevent against infiltration campaigns. Furthermore, they [14] concluded that socialbots could be profitable and could cause serious privacy breaches. Therefore, socially-aware software security could be at risk.

Threats. As socialbots infiltrate social campaigns, they could pose some security threats [15]. By polluting the social relationship in OSN, the polluted OSN can no longer be trusted. Socialbots could be used to spread rumors, spread malware, influence trading by giving fake high rates to online products. Furthermore, they could perform online surveillance and harvest users private data to use them for targeted SPAM or phishing campaigns.

Detection. As botnets on OSNs are relatively new, their detection mechanisms are not mature enough. Therefore, some OSN mechanisms are mentioned here instead of Section 4.

Kartaltepe et al. [39] proposed a mechanism to detect botnets that use OSNs for C&C. Their proposal has server-side and client-side countermeasure. The server-side detections is based on the fact that posts are expected to be in plain test. It looks for text attributes and uses a light-weight machine learning algorithm for real-time detection. It also follows any URL in the post to make sure it is from trusted sources. Furthermore, to determine if a process is a bot, the client-side looks for three attributes, namely; self-concealing, dubious network traffic, and unreliable provenance. To determine if a process is self-concealing, they check if it was started without human interaction and it does not have a graphical user interface. Dubious network traffic processes can be detected if they have exclusive requests to social network, encode

[4]CAPTCHA: an automated challenge-response tests to ensure that the response is generated by a human

their text, or download suspicious files (executable, compressed, library). Unreliable provenance processes are processes that do not have reliable origins. These processes can be determined if they are self-reference replication, have dynamic code injection feature, or do not have digital signature.

In [21], Chu et al. proposed a classification system to determine whether tweets on Twitter belong to a human, bot, or cyborg[5]. They studied over 500,000 accounts to find the difference between human, bots, and cyborg in tweeting content and behavior. Their classifier is comprised of the following four components. (1) **Entropy Component**: to detect the regularity and periods of users' tweets. (2) **Machine Learning Component**: to detect spam tweets. (3) **Account Properties Component**: to help identify bots by checking external URL ratio in the tweets. Checking the tweet device (web, mobile, or API) helps in detection bots. (4) **Decision Maker Component**: uses the input of the previous three components to determine if the user is human, bot, or cyborg.

4 Botnet Detection and Defence

Early botnet detection methods were designed to detect botnets using their signatures [1]. Such systems cannot detect unknown botnets. Therefore, signature-based detections become available too late, after a botnet has done its initial damage. However, these detection methods are useful to avoid infection by the same old malware. Daniel et al. [53] classified botnet detection methods into passive and active detections while Trend Micro's report [2] suggests that observing the botnet behavior is an important stage in detecting botnets. It [2] divided botnets' observable behavior into three types, which are as follows:

- **Network Based Behavior**: Botmasters, while communicating with their bots (using IRC, P2P or HTTP C&C), generate observable network traffic. This traffic can be used to detect individual bots and their C&C servers. Many botnet use dynamic DNS to locate their C&C server. Therefore, abnormal DNS queries may be used to detect botnets.
- **Host Based Behavior**: While compromising computers, botnets make sequence of system/library calls (e.g. modifying system registries and/or files, creating network connections and/or disabling antivirus programs). The sequences of system/library calls made by botnets are observable for their detection.
- **Global Correlated Behavior**: The fundamental structures and mechanisms of botnets give a global behavioral characteristics, which are unlikely to change until fundamental structure and mechanism of botnets is not changed. Therefore these global observable behaviors are most valuable to detect botnets.

[5]Cyborg: is either bot-assisted human or human-assisted bot.

In this section, we discuss botnet detection methods available in the literature which do not require signatures and are capable of detecting unknown botnets. These methods are categorized into the following three main categories, provided from the most common to the least common detection methods; (1) Botnet behavior-based detection (Section 4.1): This is the most common technique used to detect botnets based on their abnormal traffic behavior, whether bot generated traffic, or bot DNS queries. The bulk of the surveyed detection methods fall under this section. (2) Botmaster traceback detection (Section 4.2): This is a less common detection method based on tracing botmasters during botnet attacks or when bots report to their botmasters. (3) Detection using virtual machines (Section 4.3): This is an expensive approach based on running virtual machines on hosts in order to detect botnets.

The reader can refer to Table 1 for comparison between various botnet detection methods. This table (will be discussed later) highlights important features of different botnet detections like; the ability to detect encrypted bots, protocol and structure independency, real-time detection ability, computational cost, etc.

4.1 Botnet Behavior Detection

Detecting botnets based on their traffic behavior is further classified into three subsections: (1) C&C Traffic Behavior: to detect abnormality in C&C traffic (C&C communication channel). (2) Bot Generated Traffic: to detect abnormality in the traffic generated because of the botnet (e.g. SPAM, scan, DDoSA, etc). (3) DNS Traffic: to detect abnormality in DNS traffic caused by the botnet.

4.1.1 C&C Traffic

Based on few IRC attributes, Mazzariello [45] modelled IRC user behavior. The author's target was to separate human user generated traffic from automated IRC traffic using language complexity, vocabulary and response times. Support Vector Machine (SVM) [66] and J48 [55] decision trees were used in the experiment. Though the experiment was a success, it was not clear if this was due to the algorithm or the dataset used.

Strayer et al. [62] used filters in pipeline manner to separate botnet traffic. The filtered botnet traffic flows are classified into IRC and non-IRC flows. Then, the algorithm looks for relationships between these flows in the correlation stage. Finally, the Topological Analysis stage takes place in three steps. First, looking for common endpoints by examining clusters IP addresses. Second, correlating traffic clusters by locating traffic in other flows that share the same endpoint. Third, flows are examined to find out which one is between the botmaster and the endpoint.

The authors in [44] propose a Machine Learning (ML) technique to detect IRC-based botnet C&C traffic. After filtering out non-IRC traffic, they tried to identify C&C

hosts by isolating the flows that likely contain C&C traffic, and by correlating them to group flows that belong to the same botnet. To reduce the flow size all port scanning traffic (i.e. TCP Syn or TCP Rst) is eliminated. To avoid software update and rich web page transfer traffic, high bandwidth traffic flows are eliminated too. Furthermore, all short lived flows (few packets or seconds) are eliminated because they can not belong to botnets. This method could result in high false positive rates and could impose considerable computational overhead.

Balram and Wilscy [9] propose a bot detection system for a single host such as PCs, which are vulnerable to phishing, data stealing and data exfiltration. The system filters out the normal traffic generated on the host and analyses the remaining suspicious traffic. Their results suggest that their real time detections system can achieve high detection rate and low false positive rate.

4.1.2 Bot Generated Traffic

Binkley et al. [12] tried to detect IRC botnets based on traffic anomaly. They considered an IRC channel to be malicious if most of its hosts are performing TCP SYN scanning. They collected three tuples for their analysis; (1) TCP SYN scanner to determine types of scanning on the network, (2) IRC channel list to determine IRC channel name and IRC hosts in the channel, (3) IRC node list to determine any IP address that belongs to any IRC channel. Using these tuples, they were able to generate reports of malicious channels, sort IRC channels by maximum number of messages, analyze host statistics of IRC channels, record IRC servers, etc. This algorithm is not signature-based and should work with unknown IRC botnets, but it cannot detect encrypted botnets.

Akiyama et al. [5] suggested that bots of the same botnet have regularities in *relationship*, *response* and *synchronization* and used these measures for botnet detection. Since all bots take commands from the botmaster, there is a one-to-many *relationship* -between the bots and their botmaster- even if there is no direct connection over a single layer. In addition, when bots receive command from the botmaster, they *respond* automatically and without mistakes. This is very different from human responses while chatting. Furthermore, when bots receive a command, they take the same action almost at the same time. For example, when the botmaster sends commands for DDoS attack, all participating bots start the attack at the same time. This *synchronization* is used as a detection metric measure. This detection method could falsely identify high-demand legitimate nodes as botmasters. Furthermore, in order to avoid detection, botnets could adjust their response time to something similar to human response.

4.1.3 DNS Traffic

Botmaster use DNS rallying to make their botnets invisible and portable. Choi et al. [20] proposed botnet detection mechanism by monitoring their DNS traffic. According to the authors, bots use DNS queries either to connect or to migrate to another C&C server. The DNS traffic has a unique feature that they define as group activity. Bots can be detected by using the group activity property of botnet DNS traffic while bots are connecting to their server or migrating to another server. There are three factors that help in distinguishing botnet DNS queries from legitimate DNS queries [20]; (1) queries to C&C servers come only from botnet members (fixed IP address space size), (2) botnet members migrate and act at the same time, which leads to temporary and synchronized DNS queries, (3) botnets usually use DDNS for C&C servers.

For a botmaster to keep its bot hidden and portable, it relies on DNS to rally infected hosts. In botnets, DNS queries can appear for many reasons. They appear during rallying process after infection, during malicious activities like spam or DoS attacks, during C&C server migration, during C&C server IP address change, or after C&C server or network link failure. Based on the aforementioned five situations of DNS query used in botnets, the authors have developed a **Botnet DNS Q Detection** algorithms, which distinguishes the botnet. This algorithm starts by building a database for DNS queries comprised of the source IP address, domain name and timestamp. Then, they group DNS query data using the domain name and timestamp field. After that, they remove redundant DNS queries. Finally, botnet DNS queries are detected using a numerically computed some similarity factor [20] This algorithm cannot detect botnets migrating to another C&C server. Therefore, they developed a **Migrating Botnet Detection** algorithm by modifying the botnet DNS query detection algorithm. Similarly, this algorithm starts by building a database for DNS queries comprised of the source IP address, domain name and timestamp. Then, it groups DNS query data using the domain name and timestamp field. After that, it removes redundant DNS queries. The next step will be to compare IP lists of different domain name with same size of IP list, because bots use two different domain names for the C&C server during migration [20].

These algorithms are protocol and structure independent and are capable of detecting unknown and encrypted botnets. However, these are not for real-time detections and have low accuracy for small networks. Furthermore, they are very sensitive to threshold values which need to be chosen very carefully to balance false positives and false negative rates.

4.2 Botmaster Traceback Detection

Most of the research on botnets focuses on detection and removal of C&C servers and bots in a network [57]. Detection of botmasters is not addressed as often because it is a more challenging task. Botmasters do not need to stay online for long periods of time. As soon as they give their command(s), they can go offline and leave the hardwork to their bots. Therefore, traceback of botmasters

Table 1: Botnet detection methods comparison

No.	Reference	Category	Note	Encrypted Bot	Protocol Independent	Structure Independent	Real Time	Low False +ve/-ve	Active System	Low Cost
1	[45]	Behavior (CCT)	Mining (IRC)					✓		✓
2	[62]	Behavior (CCT)	Honeynets (IRC)				✓			✓
3	[44]	Behavior (CCT)	ML (IRC)							
4	[9]	Behavior (CCT)	ML (HTTP)				✓	✓		∼
5	[12]	Behavior (BGT)	Anomaly					∼		∼
6	[5]	Behavior (BGT)	Anomaly							✓
7	[20]	Behavior (DNS)	Anomaly	✓	✓	✓		∼		
8	[57]	Trace-back	∼	✓	✓		✓	✓	✓	✓
9	[18]	Trace-back	∼				✓	✓	✓	✓
10	[43]	Virtual Machine	BotTracer	✓	✓	✓	✓	✓		
11	[29]	System Example	Rishi							✓
12	[32]	System Example	BotHunter		✓	✓	✓	✓		∼
13	[33]	System Example	BotSniffer	✓				✓		∼
14	[31]	System Example	BotMiner	✓	✓	✓		✓		

∼: Comment or Data are Not Available ✓: The Algorithm Has This Advantage Bhv: Behavior
CCT: C&C Traffic BGT: Bot Generated Traffic TB: Trace-back
ML: Machine Learning VM: Virtual Machine SysEx: System Example

need to be carried out in real-time. Furthermore, botmaster usually connect to their bots via stepping stones in order to hide themselves. Botmaster's C&C traffic is always low-volume and botmaster may hide it even more using encryption [57]. Ramsbrock et al. [57] proposed a unique real-time watermarking botmaster traceback technique that is resilient to encryption and stepping stones. They assumed that their tracer is in control of a bot which is capable of responding to the botmaster. Their approach depends on this bot node injecting watermark when it responses to the botmaster. The watermarking is applied as follows: (1) Random packet pairs are selected. (2) The length of these packets are adjusted by padding in a way that the length difference in each packet pair falls into a predefined range. (3) For encrypted botnet traffic, they developed a hybrid length-timing watermarking method in which the watermarking packet need to be sent at specific time. For their hybrid length-timing watermarking method to work, the assumption that network jitter is limited and knowledge of the availability time of each watermarking packet must hold.

According to Chi et al. [18], once bots receive an attack command, they attack the victim at the same time. So, they proposed a method to detect the botmaster during an attack starting from the victim and working backwards through network nodes. During this detection process, the malicious traffic is blocked router by router. Their work is based on the assumption that routers from the botmaster to the victim are fixed during a given timeframe. It is also assumed that these routers are not compromised. When the IDS that is installed on the victim detects an attack, it sends diagnose request to its edge router setting the TTL to 255. The router starts a marking mode on its interfaces and notifies the victim that it has started the marking mode. As a result, all packets coming to the victim will have their hop-count equals to zero and ID equals the ID of the router's interface that processed the packet. Now the victim sends specific diagnose request to the router's interface that processed the suspicious packets. This process is repeated till the botmaster's router is reached, and the botmaster is detected [18]. This is a real-time detection algorithm that should be capable of detecting unknown botnets. It has low computational power and low false negative rates. However, this algorithm cannot detect encrypted botnets and is designed for IRC-based botnets.

4.3 Detection Using Virtual Machine

Liu et al. [43] proposed BotTracer, a detection technique that is based on virtual machine analysis of program executions. This technique is based on the assumption that bots should have three main features; (1) the bot program starts automatically without user intervention, (2) the bot must start C&C communication, (3) the bot must launch an attack. The BotTracer begins by starting a virtual machine (on the same host) that has identical image of the host system when it starts. This virtual machine will have all autostart processes on the original host but it will be free from any human interaction. Then, the BotTracer will monitor all these processes' automatic communications to detect C&C communications. Finally, BotTracer monitors the processes -that initiated suspected C&C communication- for all system-level activities and traffic patterns. Therefore, once a bot starts malicious activity, it will be detected. This is a real-time technique that is capable of detecting unknown bots regardless of their protocol, with low false positive rate, even if the C&C traffic is encrypted. However, BotTracer has high computational requirement hence virtual machine will degrade the user performance. The BotTracer will not protect against zero day attacks where the bot stay inactive waiting for a specific date and time. Furthermore, for many bots that check for virtual machine presence, the BotTracer will not work.

4.4 Examples of Botnet Detection Systems

Botnet detection system usually use more than one detection approach. For example, a detection system could use signature, C&C and botnet generated traffic to detect botnets. Therefore, it is not feasible to put these detection systems under one classification.

1) **Rishi** [29]: this is an IRC-based botnet detection system that uses IRC channel names for detection. It monitors the network traffic for suspicious IRC channel names. Rishi starts by filtering all TCP packets containing IRC-related headers. These packets are identified by any of these keywords; NICK, JOIN, USER, QUIT and MODE. Then the following information is extracted from the captured packets; connection time, source port and IP address, destination port and IP address, IRC channel and IRC nickname. After that, nicknames are passed to the analyzer where they are scored. Higher scores reflect higher probability of botnet connections. Connections with scores higher than a preset threshold are marked as suspicious and a warning email is generated and sent to the network administrator.
2) **BotHunter** [32]: this is a botnet detection system that is based on a predefined botnet infection lifecycle. This system works in real time and can detect bots regardless of the network protocol or C&C structure as long as the botnet's behavior follows a predefined infection cycle dialog model (i.e. target scanning, infection exploitation, botnet binary downloading, botnet code execution, C&C communication and outbound scanning). BotHunter is comprised of three engines; **S**tatistical s**C**an **A**nomaly **D**etection **E**ngine (SCADE), **S**tatistical pay**L**oad **A**nomaly **D**etection **E**ngine (SLADE) and Signature Engine. SCADE is responsible for the detection of inbound and outbound scan activities. SLADE detects abnormalities in byte-distribution payloads. The signature engine is capable of detecting dialog warnings from a predefined botnet infection warning model. Furthermore, BotHunter uses a correlator to evaluate all messages (dialogs) from the anomaly detection engines (SCADE and SLADE).
3) **BotSniffer** [33]: this is a botnet detection system that is based on traffic anomaly in Local Area Networks (LANs). It is based on the assumption that all the bots respond to a command in crowds and in the same way. It looks for similarities in botnet's traffic spatial-temporal correlations. The BotSniffer algorithm is comprised of two main blocks, monitor engine and correlation engine. The monitor engine is made up of three parts; (1) Preprocessing: to reduce traffic volume using filters and whitelists. (2) C&C-like protocol matcher: to collect suspicious IRC and HTTP traffic using port-independent protocol matcher. (3) Response Detector: to detect abnormally-high scan rates, weighted failed connection rate, MX DNS query and SMTP connections. The correlation engine runs in three phases; (1) Grouping: performing 2-tuple (destination IP and port number) grouping of the nodes. (2) Groups analysis: performing Response-Crowd-Density-Check algorithm, utilizing sequential probability ratio testing, to check for dense response crowds within the groups. It also performs Response-Crowd-Density-Check algorithm looking for crowds with similar responses. (3) Botnet Alert: to issue an alert if any suspicious spatial-temporal correlation C&C is detected.
4) **BotMiner** [31]: this is a botnet detection system that is based on a framework made of three main phases; monitoring, clustering, and correlating. First, in the monitoring phase, two monitoring engines -namely C&C communication traffic engine (C-plane), and activity engine (A-plane)- are used. Each engine keeps logs of its traffic analysis. The C-plane monitors both TCP and UDP flows to determine who is talking to whom. The A-plane monitors network activities to determine who is doing what (e.g. scan, spam) by detecting abnormally-high scan rates or weighted failed connection rate. Second, in the clustering phase, the C-plane clustering is performed by looking for clusters of hosts that share same communication patterns. These clusters are victimized by calculating four random variables, namely; number of flows per hour, number of packets per hour, average number of bytes per packet and average number of bytes per second. In A-plane clustering, hosts are first clustered based on their malicious activities (e.g. scanning) then are clustered based on activity features (e.g. port number). Finally, a cross-plane correlation is performed to find intersection between the two clusters in the previous phase. The intersection means that these hosts are part of a botnet.

To summarize, as Table 1 shows, though Rishi is a low cost botnet detection system, it is a non-real-time passive system that can only detect un-encrypted IRC botnets. The other three detection systems (BotHunter, BotSniffer and BotMiner) are proposed by Gu et al. BotHunter is a real-time, protocol and structure independent detection system capable of detecting unknown botnets with few false positives/negatives. However, it is a passive system that requires botnet to follow a predefined infection cycle dialog model to be detected and it is not capable of detecting encrypted botnets. BotSniffer is capable of detecting encrypted botnets with low false positives/negatives rates, but it is protocol and structure dependent and is not a real-time system and works for LANs only. Finally, BotMiner is a passive non-real-time system. It is a low cost detection system that is capable of detecting encrypted botnets regardless of their protocol or structure with low false positives/negatives rates.

4.5 Detection Methods Summary

To summarize, in this section, we discussed botnet detection methods. These methods are categorized into three categories namely; botnet behavior-based detection, botmaster traceback detection, and detection using virtual machines.

As Table 1 shows, most of behavior-based detection methods (except DNS traffic analysis detections) are protocol dependent, cannot detect encrypted botnets and are neither real-time nor active methods. However, most of them have acceptable false positive/negative rates and acceptable computational cost. DNS traffic analysis detections are capable of detecting encrypted bots, regardless of their protocol.

Traceback detection methods are real-time active techniques that have acceptable false positive/negative rates and acceptable computational cost, but they are not structure independent.

Detection using virtual machines seems to be working for encrypted bots regardless of their protocol or structure. It is a real-time algorithm with acceptable false positive/negative rates. However this system is passive and has high computational overhead.

4.6 Defence and Post-Detection Reactions

According to [36], once a botnet is detected, it needs to be tracked and brought down. First, a copy of the bot needs to be analyzed to understand the bot behavior. To get a copy of the bot, the analyzer needs to use methods similar to honeypots. After that, the bots code needs to be studied to find out; how the communication is done within the botnet, how does new members join the botnet, and find the whereabouts of the botmaster. Finally, the source of the bot is brought down (physically) by the authorities [36].

Very few papers proposed post-detection procedures against botnet. Vogt et al. [67] suggested that superbotnets must be examined by the research community, so that defences against this threat can be developed proactively. They pointed out some weak aspects of C&C mechanism that are exhibited by traditional botnet and suggest defenders to target these weaknesses. They concluded that there are five goals that defenders could take into account to build a defence mechanism against botnets:

1) Locate or identify the adversary: At the time the adversary issues commands through the botnets' C&C, it becomes vulnerable to detection.
2) Reveal all the infected machines: If bots are pooling for botnets' commands from a known location, this polling activity can be used to reveal infected machines.
3) Command the botnet: Once the defender is familiar with the botnets' commands, (s)he can send a command to the botnet to shut it down.
4) Disable the botnet: The botnet could by paralysed by shutting down its C&C channel.
5) Disrupt Botnet Commands: By changing few bits in the adversary's commands is sufficient to disrupt adversary's control of the botnet.

5 Conclusion

Despite the fact that our knowledge about botnets is incomplete; botnets are one of the most serious threats to network security. This survey was conducted to better understand botnets and is an attempt to organize the enormous background available in this area to help researchers who are starting in this area.

In this survey, we explained botnets C&C communication, infection behaviors and models. This survey discussed some of the botnets facilitator services. Fast-Flux service networks were illustrated in great details and botnets' C&C rallying mechanisms were surveyed.

We classified botnets -based on their underlying C&C protocol- to IRC, HTTP, POP3, and P2P botnets. As a new emerging malware, social and mobile botnets' threats and potential were discussed in this survey. As mobile phones with networking capabilities have become more affordable, the threat of mobile botnets have increased. Mobile botnets could spread through SMS or MMS services. Their effect could be very damaging as the security measures against mobile botnets may not have been designed for mobile device.

Furthermore, botnet detection methods are surveyed in detail. Detection methods have been classified into three classes. First, behavior-based detection where botnets are detected using; C&C traffic behavior, bot generated traffic behavior, or DNS traffic behavior. Second, botmaster traceback detection is described. Then, a virtual machine detection method is explained. Finally, examples of botnets detection systems were explained (i.e. *Rishi*, *BotHunter*, *BotSniffer* and *BotMiner*). The survey is concluded with the botnets defence measures that should be taken after detecting a botnet.

Acknowledgments

M. Mahmoud acknowledges funding from King Fahd University of Petroleum & Minerals (KFUPM). This work was done while M. Mahmoud was doing his Ph.D. at Carleton University. M. Nir and A. Matrawy acknowledge funding from Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] "SNORT," Mar. 2006. (https://www.snort.org/)
[2] "Taxonomy of botnet threats," white paper, Trend Micro Incorporated, Nov. 2006. (http://www.cs.ucsb.edu/ kemm/courses/cs595G/TM06.pdf)

[3] "Emerging cyber threats," Technical Report, Georgia Tech. Information Security Center, Oct. 2008. (https://www.gtisc.gatech.edu/pdf/Threats_Report_2008.pdf)

[4] "The möbius tool," Apr. 2011. (https://www.mobius.illinois.edu/)

[5] M. Akiyama, T. Kawamoto, M. Shimamura, T. Yokoyama, Y. Kadobayashi, and S. Yamaguchi, "A proposal of metrics for botnet detection based on its cooperative behavior," in *International Symposium on Applications and the Internet Workshops*, pp. 82–82, Jan. 2007.

[6] I. Asrar, "Could Sexy Space be the Birth of the SMS Botnet?", July 2009. (http://www.symantec.com/connect/blogs/could-sexy-space-be-birth-sms-botnet)

[7] P. Bächer, T. Holz, M. Kötter, and G. Wicherski, "Know your Enemy: Tracking Botnets," Oct. 2008. (http://www.honeynet.org/papers/bots)

[8] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A survey of botnet technology and defenses," in *Cybersecurity Applications Technology Conference For Homeland Security*, pp. 299–304, Mar. 2009.

[9] S. Balram and M. Wilscy, "User traffic profile for traffic reduction and effective bot c&c detection," *International Journal of Network Security*, vol. 16, pp. 46–52, Jan. 2014.

[10] J. Baltazar, J. Costoya, and R. Flores, "The real face of koobface: The largest web 2.0 botnet explained," Technical Report, Trend Micro Incorporated, July 2009.

[11] P. Barford and V. Yegneswaran, "An inside look at botnets," in *Advances in Information Security*, vol. 27, pp. 171–191, Springer, Mar. 2007.

[12] J. R. Binkley and S. Singh, "An algorithm for anomaly-based botnet detection," in *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet*, pp. 7–7, Berkeley, CA, USA, 2006.

[13] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "The socialbot network: When bots socialize for fame and money," in *Proceedings of the ACM 27th Annual Computer Security Applications Conference*, pp. 93–102, New York, NY, USA, 2011.

[14] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "Design and analysis of a social botnet," *Computer Networks*, vol. 57, no. 2, pp. 556–578, Feb. 2013.

[15] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "Key challenges in defending against malicious socialbots," in *Proceedings of the 5th USENIX conference on Large-Scale Exploits and Emergent Threats (LEET'12)*, pp. 12–12, Berkeley, CA, USA, 2012.

[16] A. Caglayan, M. Toothaker, D. Drapeau, D. Burke, and G. Eaton, "Real-time detection of fast flux service networks," in *Proceedings of the IEEE Cybersecurity Applications & Technology Conference for Homeland Security*, pp. 285–292, Washington, DC, USA, Mar. 2009.

[17] J. A. Chandler, "Liability for Botnet Attack," *Canadian Journal of Law and Technology*, vol. 5, pp. 13–25, Mar. 2006.

[18] Z. Chi and Z. Zhao, "Detecting and blocking malicious traffic caused by IRC protocol based botnets," in *IFIP International Conference on Network and Parallel Computing*, pp. 485–489, Dalian, China, Sep. 2007.

[19] K. Chiang and L. Lloyd, "A case study of the rustock rootkit and spam bot," in *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, pp. 10–10, Berkeley, CA, USA, 2007. Association.

[20] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet detection by monitoring group activities in dns traffic," in *The 7th IEEE International Conference on Computer and Information Technology*, pp. 715–720, Oct. 2007.

[21] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who is tweeting on twitter: Human, not, or cyborg?" in *Proceedings of the ACM 26th Annual Computer Security Applications Conference (ACSAC'10)*, pp. 21–30, New York, NY, USA, 2010.

[22] Z. Coburn and G. Marra, "Real boy: Believable twitter bots," July 2012.

[23] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets," in *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*, pp. 6–6, Berkeley, CA, USA, 2005.

[24] X. Cui, B. Fang, L. Yin, X. Liu, and T. Zang, "Andbot: Towards Advanced Mobile Botnets," in *Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats (LEET'11)*, pp. 11–11, Berkeley, CA, USA, 2011.

[25] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *Proceedings of the 13 th Network and Distributed System Security Symposium NDSS*, 2006.

[26] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," pp. 268–273, Los Alamitos, CA, USA, June 2009.

[27] D. Fisher, "Storm, nugache lead dangerous new botnet barrage," Dec. 2007. Online Article.

[28] A. R. Flø and A. Jøsang, "Consequences of Botnets Spreading to Mobile Devices," in *Proceedings of the 14th Nordic Conference on Secure IT Systems (NordSec 2009)*, Oct. 2009.

[29] J. Goebel and T. Holz, "Rishi: identify bot contaminated hosts by IRC nickname evaluation," in *Proceedings of the first Conference on First Workshop on Hot Topics in Understanding Botnets*, pp. 8, Berkeley, CA, USA, 2007.

[30] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon, "Peer-to-peer botnets: Overview and

case study," in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pp. 1, Berkeley, CA, USA, 2007.

[31] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proceedings of the 17th Conference on Security Symposium*, pp. 139–154, Berkeley, CA, USA, 2008.

[32] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "Bothunter: detecting malware infection through ids-driven dialog correlation," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pp. 1–16, Berkeley, CA, USA, 2007.

[33] G. Gu, J. Zhang, and W. Lee, "Botsniffer: Detecting botnet command and control channels in network traffic," in *Proceedings of 16th Annual Network and Distributed System Security Symposium (NDSS'08)*, Reston, VA, USA, February 2008.

[34] K. J. Higgins, "Smartphone weather app builds a mobile botnet," Mar. 2010.

[35] T. Holz, C. Gorecki, K. Rieck, and F. Freiling, "Measuring and detecting fast-flux service networks," in *The 15th Network and Distributed System Security Symposium (NDSS'08)*, Reston, VA, USA, Feb. 2008.

[36] T. Holz, M. Steiner, F. Dahl, E. biersack, and F. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pp. 1–9, Berkeley, CA, USA, 2008.

[37] T. Hwang, I. Pearce, and M. Nanis, "Socialbots: Voices from the fronts," *interactions*, vol. 19, pp. 38–45, Mar. 2012.

[38] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, pp. 7–7, Berkeley, CA, USA, Apr. 2007.

[39] E. Kartaltepe, J. Morales, S. Xu, and R. Sandhu, "Social network-based botnet command-and-control: Emerging threats and countermeasures," in *Applied Cryptography and Network Security*, LNCS 6123, pp. 511–528, Springer-Verlag, 2010.

[40] C. Li, W. Jiang, and X. Zou, "Botnet: survey and case study," in *Fourth International Conference on Innovative Computing, Information and Control*, pp. 1184–1187, Dec. 2009.

[41] C. Y. Liu, C. H. Peng, and I. C. Lin, "A survey of botnet architecture and botnet detectection techniques," *International Journal of Network Security*, vol. 16, no. 2, pp. 81–89, Mar. 2014.

[42] J. Liu, Y. Xiao, K. Ghaboosi, H. Deng, and J. Zhang, "Botnet: classification, attacks, detection, tracing, and preventive measures," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, pp. 11, 2009.

[43] L. Liu, S. Chen, G. Yan, and Z. Zhang, "Bottracer: Execution-based bot-like malware detection," in *Information Security*, pp. 97–113, 2008.

[44] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Using machine learning technliques to identify botnet traffic," in *The 31st IEEE Conference on Local Computer Networks*, pp. 967–974, Nov. 2006.

[45] C. Mazzariello, "IRC traffic analysis for botnet detection," in *Fourth International Conference on Information Assurance and Security (ISIAS'08)*, pp. 318–323, Naples, Italy, Sep. 2008.

[46] S. Nagaraja, A. Houmansadr, P. Piyawongwisal, V. Singh, P. Agarwal, and N. Borisov, "Stegobot: A covert social network botnet," in *Information Hiding*, LNCS 6958, pp. 299–313, Springer-Verlag, 2011.

[47] J. Nazario, "Blackenergy ddos bot analysis," Technical Report, Arbor Networks, Oct. 2007.

[48] J. Nazario, "Twitter-based botnet command channel," Aug. 2009.

[49] J. Nazario and T. Holz, "As the net churns: Fast-flux botnet observations," in *The 3rd International Conference on Malicious and Unwanted Software*, pp. 24–31, Oct. 2008.

[50] J. Oikarinen and D. Reed, "Internet relay chat protocol," RFC 1459, May 1993.

[51] M. Overton, "Bots and botnets: risks, issues and prevention," in *proceedings of virus bulletin Conference*, Virus Bulletin, Oct. 2005.

[52] E. Passerini, R. Plaeari, L. Martignoni, and D. Bruschi, "Fluxor: Detecting and monitoring fast-flux service networks," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, LNCS 5137, pp. 186–206, July 2008.

[53] D. Plohmann, E. Gerhards-Padilla, and F. Leder, "Botnets: measurement, detection, disinfection and defence," in *ENISA Workshop* (Giles Hogben, ed.), Mar. 2011.

[54] P. Porras, H. Saidi, and V. Yegneswaran, "An analysis of the ikee.b (duh) iphone botnet," Technical Report, SRI International, CA, 94025, USA, Dec. 2009.

[55] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., 1993.

[56] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC'06)*, pp. 41–52, New York, NY, USA, Oct. 2006.

[57] D. Ramsbrock, X. Wang, and X. Jiang, "A first step towards live botmaster traceback," in *Recent Advances in Intrusion Detection*, LNCS 5230, pp. 59–77, Springer-Verlag, 2008.

[58] E. Van Ruitenbeek and W. H. Sanders, "Modeling peer-to-peer botnets," in *Fifth International Conference on Quantitative Evaluation of Systems*, pp. 307–316, Sep. 2008.

[59] Y. H. Shin and E. G. Im, "A survey of botnet: consequences, defenses and challenges," in *The fourth Joint Workshop on Information Security (JWIS'09)*, Kaohsiung, Taiwan, Aug. 2009.

[60] K. Singh, A. Srivastava, J. Giffin, and W. Lee, "Evaluating email's feasibility for botnet command and control," in *IEEE International Conference on Dependable Systems and Networks with FTCS and DCC*, pp. 376–385, June 2008.

[61] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich, "Analysis of the storm and nugache trojans: P2p is here," *The USENIX Magazine*, vol. 32, pp. 18–27, Dec. 2007.

[62] W. T. Strayer, R. Walsh, C. Livadas, and D. Lapsley, "Detecting botnets with tight command and control," in *Proceedings the 31st IEEE Conference on Local Computer Networks*, pp. 195–202, Cambridge, MA, Nov. 2006.

[63] V. L. Thing, M. Sloman, and N. Dulay, "A survey of bots used for distributed denial of service attacks," in *New Approaches for Security, Privacy and Trust in Complex Environments*, vol. 232/2007 of *IFIP International Federation for Information Processing*, pp. 229–240, Boston, USA, Nov. 2007.

[64] P. Traynor, M. Lin, M. Ongtang, v. Rao, T. Jaeger, P. McDaniel, and T. La Porta, "On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core," in *Proceedings of the 16th ACM conference on Computer and Communications Security (CCS'09)*, pp. 223–234, New York, NY, USA, 2009.

[65] P. Traynor, P. McDaniel, and T. La Porta, "On Attack Causality on Internet-Connected Cellular Networks," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium (SS'07)*, pp. 21:1–21:16, Berkeley, CA, USA, 2007.

[66] V. Vapnik, S. E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in *Advances in Neural Information Processing Systems*, vol. 9, pp. 281–287, 1996.

[67] R. Vogt, J. Aycock, and M. J. Jacobson, "Army of botnets," in *Proceedings of Network and Distributed System Security Symposium (NDSS'07)*, pp. 111–123, Reston, VA, USA, Feb. 2007.

[68] P. Wand, S. Sparks, and C. C. Zou, "An advanced hybrid peer-to-peer botnet," in *Proceedings of the first Conference on First Workshop on Hot Topics in Understanding Botnets*, pp. 2, Berkeley, CA, USA, 2007.

[69] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts, and K. Han, "Botnet research survey," in *32nd Annual IEEE International Conference on Computer Software and Applications*, pp. 967–972, Aug. 2008.

Muhammad Mahmoud is an Assistant Professor at King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia. He received the Ph.D. degree in electrical and computer engineering from Carleton University, Ottawa, Canada. His research interests include network security, and Communication Network Protocols. His research has been supported by KFUPM.

Manjinder Nir is currently a fourth year Ph.D. candidate in the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. He received B.Tech. and M.Tech. degrees in Electronics and Communication Engineering from Punjab Technical University, Punjab, India. His research interests include computer networking and pervasive computing.

Ashraf Matrawy is an Associate Professor and the Associate Director of the School of Information Technology at Carleton University. He received the Ph.D. degree in electrical engineering from Carleton University. He is a senior member of the IEEE, serves on the editorial board of the IEEE Communications Surveys and Tutorials journal, and has served as a technical program committee member of a number of international conferences. His research interests include reliable and secure computer networking. His research has been supported by CFI/ORF, NSERC, OCE, Alcatel-Lucent Canada, and Solana Networks.

Efficient Compression-Jointed Quality Controllable Scrambling Method for H.264/SVC

Ci-Lin Li[1], Chih-Yang Lin[2], and Tzung-Her Chen[1]
(Corresponding author: Tzung-Her Chen)

Dept. of Computer Science & Information Engineering, National Chiayi University[1]
No.300 Syuefu Rd., Chiayi City 60004, Taiwan
Dept. of Computer Science & Information Engineering, Asia University[2]
No.500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
(E-mail: thchen@mail.ncyu.edu.tw)

(Received Sep. 27, 2013; revised and accepted Oct. 6, 2013 & Feb. 10, 2014)

Abstract

H.264/Scalable video coding (SVC) is an ongoing standard that supports adaptation of a heterogeneous network by spatial, temporal, and quality scalability. Although H.264/SVC has drawn much attention in academia and industry, there has been little research related to H.264/SVC scrambling reported in the literature. In this study, a quality controllable scrambling scheme is proposed for H.264/SVC, in which the H.264/SVC compression processes are skillfully scrambled. Thus, the users can decide the quality of the scrambled video for different applications. In addition, the proposed scheme shows that the computation and compression overheads are negligible since only the sensitive data, including motion vector difference, the intra prediction mode, and the residual coefficients, are encrypted. The experimental results demonstrate the feasibility of the proposed scheme by four requirements: low bit-rate, low computational complexity, format compliance, and preservation of video quality.

Keywords: Compression-jointed encryption, quality control, scalable video coding (SVC), video encryption

1 Introduction

With the development of network technologies and the popularity of personal computers, multimedia has become the most common data type distributed over the Internet or saved in storage devices. Thanks to the inexpensive and widely used webcams and digital cameras, multimedia data can be generated more easily. However, the amount of data generated by multimedia devices is usually large and thus inconvenient to transmit and store. Consequently, many video compression techniques have been proposed, including MPEG 1/2/4, H.261/3/AVC/SVC for various applications and services.

Although the rapid-growth of networks makes life more convenient, it also results in an increase in security concerns. For example, if a user transmits a valuable video via the Internet, malicious hackers may easily download and distribute it to anyone without authorization, seriously damaging the owner's intellectual property rights (IPR). Therefore there is a requirement for a combination of compression and encryption to enhance the security for data transmission. To achieve this, a video scrambling method is a pivotal feasible technology for video content protection.

1.1 Perceptual Security

The level of perceptual security requirement usually varies with different applications and can be classified in two categories [11, 12, 15, 22, 23], perceptual scrambling and non-perceptual scrambling. In perceptual scrambling [6], the scrambled video is a low-quality version, but the content of encrypted video is still recognizable. Hence, the user can choose to watch high quality video for a fee or a low quality video for free. This is usually used in a DRM system. In a non-perceptual scrambling method, any useful information from the video cannot be revealed to an unauthorized viewer, implying that no visible information is able to be recognized by viewing the encrypted video. In this paper, we provide a scrambling method that can flexibly control the video quality. Hence, the user can decide between perceptual or non-perceptual scrambled video.

1.2 Requirements of Video Encryption

Nowadays there are many traditional cryptography systems that provide powerfully security encryption methods. However, most of them are not appropriate for video since the amount of video data is much greater than text, for which the traditional cryptography systems were originally designed. Performing traditional encryption methods, like DES and AES requires many computations for both encryption and decryption, especially when the data size is large. In order to satisfy practical applications, the

performance of video encryption methods must be evaluated from four requirements [10, 11]: security, complexity, compression overhead and format compliance.

1) Security: Security is the basic concern for a video encryption scheme. In general, a malicious user should not be able to recover the encrypted video within a limited time even when a brute-force attack is applied.

2) Complexity: The video should be able to be viewed in various devices or transmitted over the Internet. The encryption complexity is an important issue because video compression and decompression consume a lot of computational power. Therefore, a low complexity encryption and decryption scheme is required for video streaming to meet the requirements of a real-time system.

3) Compression Overhead: Video compression utilizes temporal and spatial redundancy to reduce video size. If the redundancy property is affected by the encryption process, the video will become larger, and thus more bandwidth and storage space are also required. Therefore, a good encryption method should not greatly affect the redundancy property.

4) Format Compliance: A format-compliant video means that after encryption, the encrypted video should be able to be played by a media player. If it is not format-compliant, the perceptual scrambling will fail.

2 Related Work

According to the encryption process and the application requirement, video encryption methods can be partitioned into two types [11]: compression-independent encryption and compression-jointed encryption. In compression-independent encryption methods, the raw data or the compressed data is directly encrypted by traditional ciphers, such as DES or AES. The first compression-independent encryption is raw data encryption, which performs encryption before the compression process. In 2003, Maniccam and Bourbakis [13] presented a video encryption by permuting video pixel values to encrypt the raw video data. Socek et al. [20] also permuted raw pixels to break the correlation by sorting permutation. The sorting permutation order is determined by the first frame. Since the compression standard is designed to reduce the redundancy by using a hybrid coding approach, performing encryption before the compression results in much less redundancy and higher compression rates.

Another approach is encryption after compression. In 1998, Qiao and Nahrstedt [16] proposed a video encryption algorithm on the MPEG standard. This algorithm analyzes the video bitstream to partition each byte into two parts. Then, one part is encrypted with a traditional cipher, another part performs an XOR operation with a secret key. Fan et al. [3] proposed a video encryption method after the compression process on the H.264/AVC standard. They divided the video bitstream into four parts: slice header, MB header, intra-residual data, and inter-residual data, and encrypted them with AES (Advanced Encryption Standard). Since these methods encrypt the video bitstream with traditional ciphers, they impose a high computational cost, and cannot maintain format compliance.

In the compression-jointed encryption methods, only the significant data or the sensitive information of a video, such as motion vector differences, residual coefficients, and variance length codeword, is encrypted. In other words, it is a partial encryption method. The key to this kind of method is to determine what data should be encrypted because the selected data will affect security, compression rate, and format compliance.

In 1996, Tang [21] permuted the zigzag order and Shi et al. [19] encrypted the sign values of DCT coefficients for a MPEG video. For H.264/AVC, Ahn et al. [1] proposed a digital video scramble method using an intra-prediction mode, which uses the properties of fixed length coding and a VLC table. However, Li et al. [7] pointed out that [1] only provides a limited protection and suffered from replacement attacks [8]. They also proposed an improved video encryption method for H.264/AVC by encrypting using the intra-prediction mode, inter-prediction mode, motion vector, and residual data. Hong et al. [4] analyzed the statistical behavior of the encoded H.264/AVC bitstreams, and provide the flexibility to encrypt video with difference qualities. In 2006, Lian et al. [9] also developed an encryption method using the intra-prediction mode, sign values of motion vector differences, and residual data. More research results can refer to [1, 4, 7, 9, 10, 23] using the intra prediction mode encryption, [7, 9, 10 23, 24] using the motion vector differences encryption, and [2, 7, 9, 10, 23, 24] using the residual data encryption. In short, the most common selected data encrypted are intra-prediction mode, motion vector difference, and the residual data in the compression-jointed encryption methods.

In 2007, the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) jointly extend the H.264/AVC standard to a scalable video coding (SVC) [18]. H.264/SVC is the ongoing video coding standard, and enables efficient adaptation to provide various video qualities for different devices in the heterogeneous network. H.264/SVC only needs encoding once, and then, the adaptive bitstream can be extracted for decoding, depending on the network bandwidth. H.264/SVC supports temporal scalability, spatial scalability and quality scalability, and enables decoding of partial bit streams to provide more video services. In addition, H.264/SVC achieves a high compression rate compared with previous compression methods. Hence, designing a suitable encryption/decryption method for H.264/SVC is required for practical applications.

Park and Shin [14] proposed an efficient selective encryption scheme for H.264/SVC in 2008. In this scheme, the authors encrypt the intra-prediction mode (IPM), motion vector difference (MVD), and the sign bits of the residual coefficients. Li et al. [5] proposed a layered

encryption for scalability. This encrypts using the intra-prediction mode and sign values of residual coefficients on a base layer, and encrypts the sign values of residual coefficients and MVD on the enhancement layers. Although, those schemes can encrypt the compressed video data, they did not consider the quality controllability of H.264/SVC. In other words, the quality of scrambled video cannot be controlled by a user to adapt to any kind of application.

The aim of this paper is to propose a compression-jointed quality controllable scrambling method for H.264/SVC, where the significant coefficients in the H.264/SVC compression process are skillfully scrambled, and the scalability and quality controllable, which are first proposed, can be adjusted by different applications. The proposed scheme is based on the layered scrambling method, which is suitable for digital rights management (DRM) systems. The proposed scheme is evaluated by four measurements: security, complexity, compression overhead, and format compliance.

Our goal is to achieve a low bit-rate overhand but still satisfy the property of format compliance. In fact, in the proposed scheme, each layer can be scrambled under the condition of scalability and quality control.

The rest of this paper is organized as follows. The background of H.264/SVC and compression-jointed video encryption method on H.264/SVC are described in Section 3. The experimental results and discussion are presented in Section 4. Finally, the conclusions and future work are given in Section 5.

3 Background and Compression-jointed Video Encryption Method on H.264/SVC

3.1 Overview of H.264/SVC

H.264/SVC, the latest video coding standard, enables efficient adaptation to provide various video qualities for different devices or heterogeneous networks. This coding standard is equipped with spatial scalability, temporal scalability, and SNR scalability, leading to the power of quality control and adaptation to different transmission bandwidths.

Figure 1 [18] shows the architecture of H.264/SVC with two layers. There are three scalability properties in H.264/SVC, spatial scalability, temporal scalability, and SNR scalability. The spatial scalability is achieved by spatial decimation down-sampling, where the video can be divided into several layers with different resolutions. The lowest resolution is the spatial base layer and the higher layers are the spatial enhancement layers. The temporal scalability is performed by the "hierarchical B pictures" structure [17] to decide the frame rates. Finally, the SNR scalability is achieved by the fine grain SNR scalability (FGS) or the coarse grain SNR scalability (CGS) on the entropy coding phase. In the SVC coding structure, the temporal scalability and the SNR scalability are included in each layered coding except for the spatial base layer, which is coded by the H.264/AVC compatible encoder process without considering the temporal scalability and the SNR scalability. **An example of the scalability relation is illustrated in Figure 2, where there are two spatial scalability choices, three temporal scalability choices, and two SNR scalability choic**es. In this example, the spatial decimation of QCIF includes one base layer and five enhancement layers and the CIF has six enhancement layers. Therefore, there are one base layer and eleven enhancement layers in this structure.

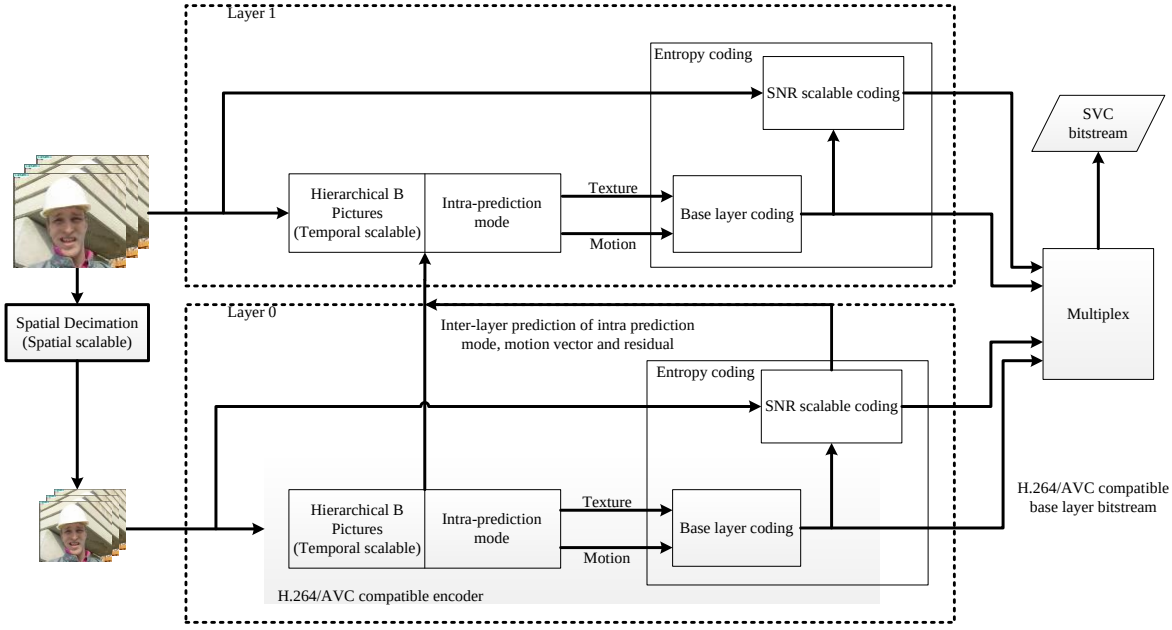


Figure 1: Architecture of the H.264/SVC general concept for spatial, temporal and SNR scalability

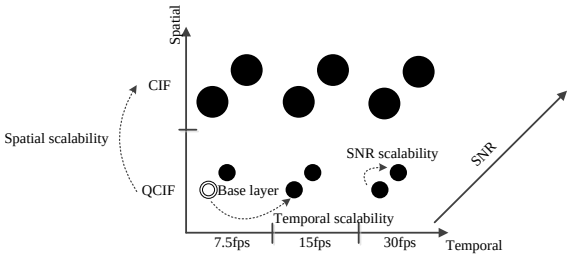


Figure 2: Layered structure of H.264/SVC

3.2 Parameter Sensitivity of H.264/SVC

In spatial scalability, the previous spatial layer has to support the inter-layer predictions to the next layer, which are MVD, IPM and the residual data. The base layer is the most important layer to support the inter-layer predictions. Therefore, for scalable video encryption, the base layer needs a greater security level than the enhancement layers, since the intra-frame contains most of the significant information of IPM and residual data [14]. However, most of the significant information of MVD of the inter-frames is maintained in the enhancement layers. Hence, the IPM, residual data, and MVD can be scrambled in difference layers for difference qualities, which means that the quality of a scrambled video can be controlled by which layer is encrypted.

3.3 The Proposed H.264/SVC Scrambling Method

3.4 The Proposed Scrambling Method

3.4.1 IPM Encryption

In H.264/SVC, there are two kinds of modes, called *prev_intra4* × *4_pred_mode* and *rem_intra4* × *4_pred_mode*, to encode nine IPMs for each luminance 4 ×4 block. The *prev_intra4*×*4_pred_mode* requiring 1 bit is the relationship between the prediction mode of the current block, called $Mod_{current}$, and the prediction mode of the neighbor block (usually using the smaller mode between the left and top block), called $Mod_{neighbor}$. If the *prev_intra*4 ×4*_pred_mode* is equal to 1, then the $Mod_{current}$ is the same as the smaller mode between the left and top block; otherwise, it needs to use *rem_intra4*×*4_pred_mode*, which provides the remaining 8 modes and requires 3 bits. The *rem_intra4*×*4_pred_mode* is defined below.

$$\begin{cases} rem\_intra4\times4\_pred\_mode = Mod_{current} - 1, & Mod_{current} > Mod_{neighbor}, \\ rem\_intra4\times4\_pred\_mode = Mod_{current}, & Mod_{current} < Mod_{neighbor}. \end{cases} \quad (1)$$

In IPM scrambling, the prediction mode is scrambled into another mode. This means that the decoding process cannot obtain the correct prediction mode, leading to the scrambling result. Here, the *rem_intra4*×*4_pred_mode* is scrambled and defined as follows.

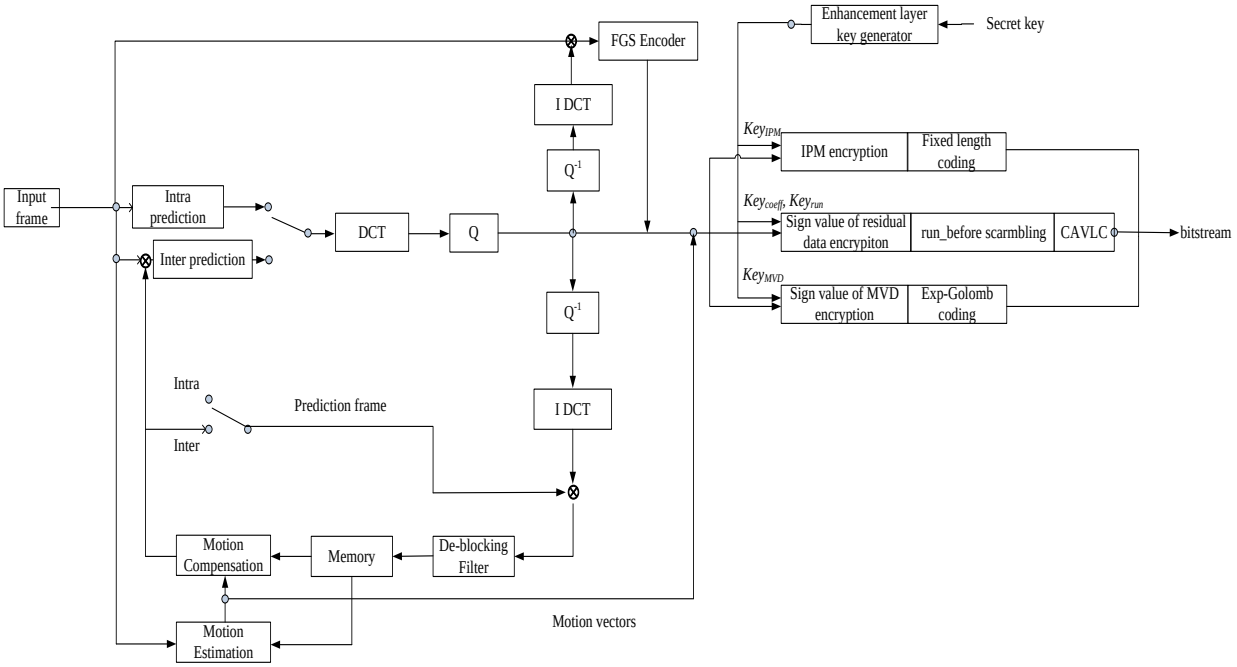


Figure 3: The flowchart of the proposed scheme for each spatial layer

According to the above analysis and requirements, the IPM, residual coefficients, and MVD are selected to scramble SVC data. The SVC joint compression and scrambling method is illustrated in Figure 3 for each spatial layer.

$$E(IPM) = rem\_intra4\times4\_pred\_mode \oplus Key_{IPM} \quad (2)$$

There is no bit expansion in the ciphered data because the IPM is scrambled by XOR operations with the fixed length of 3 bits secret key Key_{IPM}. Therefore, IPM

scrambling has low computational cost and still obeys the format compliance.

3.4.2 Residual Data Encryption

In the intra-frame, H.264/SVC makes use of IPM to remove spatial redundancy. Similarly, it utilizes MVD to remove temporal redundancy for inter-frames. The residual data, after removing the spatial or temporal redundancy, will be applied to the DCT transform and the quantization process for each 4×4 block, and then compressed by entropy coding, such as CAVLC or CABAC. If the coefficients after quantization are scrambled, the overhead of the compression rate will be increased. In order to solve this problem, the proposed scheme only scrambles the signs of non-zero residual data, and scrambles the number of zeros between two non-zero coefficients in CAVLC. The following describes the two approaches.

1) Sign Values of the Residual Data Scrambling

The key generation process is to produce the binary key Key_{coeff} to scramble the sign values of the non-zero residual data, and is defined as follows.

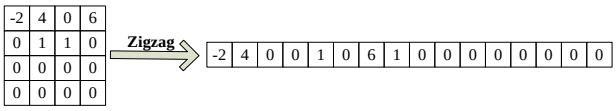
$$\begin{cases} Coeff_i = Coeff_i \times (-1), & if\ Key_{coeff} = 1 \\ Coeff_i = Coeff_i, & if\ Key_{coeff} = 0 \end{cases} \quad (3)$$

where i is the index of the zigzag order.

The sign values encryption for the residual data will degrade the video quality with a negligible computational cost, and without violation of the format compliance.

2) Zero-coefficient Scrambling

After the sign values are scrambled, the next step is to scramble the relationships between zero coefficients and non-zero coefficients. Figure 4 shows an example of the



residual coefficients of a 4×4 block.

Figure 4: The example of 16 residual coefficients of one 4×4 block

Assume that *run_before* is the number of zero coefficients between two non-zero coefficients, *TotalCoeffs* is the total number of the nonzero coefficients, and *level* represents the nonzero coefficients. According to **Figure 4**, *TotalCoeffs*=5, *level*(0)=-2, *run_before*(0)=0, *level*(1)=4, *run_before*(1)=0, *level*(2)=1, *run_before*(2)=2, *level*(3)=6, *run_before*(3)=1 and *level*(4)=1, *run_before*(4)=0. The zero-coefficient scrambling is defined as Eq. (4), where j=0,1,…,*TotalCoeffs*−1 and s_j is an element to S. The set S contains $s_1, s_2, \ldots, s_j$, which are produced by a secret Key_{run} and $s_i \neq s_j$.

$$run\_before(j) = run\_before(s_j),\ j = 0,1,...,TotalCoeffs - 1. \quad (4)$$

In **Figure 4**, if the scrambled sequence S is (2, 3, 1, 4, 0), the *run_before* becomes (2, 1, 0, 0, 0). Thus, the scrambled sequence is (0, 0, -2, 0, 4, 1, 6, 1, 0, 0, 0, 0, 0, 0, 0, 0).

3.4.3 Residual Data Encryption

In the video standard, the motion vector difference (MVD), which is the difference between the current MV and the predicted MV produced by the inter-prediction mode, is one of the important factors for video quality. Scrambling MVD can make significant changes to video content since the decoder cannot get the correct MV by adding the MVD to the previous MV. However, if the motion vector difference is changed directly, the compression rate would be harmed. In order to solve this problem, the sign values of MVD are scrambled according to the binary secret key Key_{MVD} and defined as follows.

$$\begin{cases} E(MVD) = MVD \times (-1), & if\ Key_{MVD} = 1 \\ E(MVD) = MVD, & if\ Key_{MVD} = 0 \end{cases} \quad (5)$$

In this way, the MVD is scrambled with the properties of light overhead, low computational costs, and format compliance.

3.5 Quality Controllability

According to the scalability property, the video quality is controlled by which combinations of layers are scrambled. Assume that there are α layers from the base layer to the enhancement layer α-1. The level of security is defined as follows.

3.5.1 Level 1 Security

In secure Level 1, the sign bits of non-zero residual data, the *run_before*, and the sign bits of MVD are scrambled for the enhancement layers from α-1 to α/2.

3.5.2 Level 2 Security

In secure Level 2, the sign bits of non-zero residual data, the *run_before*, and the sign bits of MVD are scrambled for the enhancement layers from α-1 to 1.

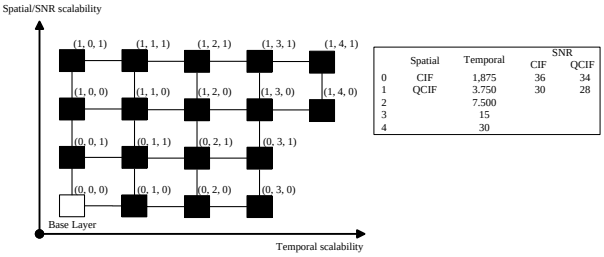
3.5.3 Level 3 Security

In secure Level 3, all layers are scrambled including the base layer. In other words, the sign bits of non-zero residual data, the *run_before*, the sign bits of MVD, and the IPM will be scrambled.

4 Experimental Classification Results and Analysis

The proposed method is implemented in an environment with Intel(R) Pentium(R) D CPU 3.40GHz and 3GB RAM under Microsoft Windows 7 and Visual C++ 2008. There are three test sequences *Bus*, *football* and *Mobile* for

evaluation. The number of coding frames is 30, and the GOP size is set to 16. The experiments perform QCIF and CIF for spatial scalability, QCIF(15fps) and CIF(30fps) for temporal scalability, and two quantization parameters for SNR scalability. Hence, the combined scalability supports a variety of spatial, temporal, and SNR scalabilities as shown in **Figure 5**. Here, α is set to 18.



	Spatial	Temporal	SNR CIF	SNR QCIF
0	CIF	1,875	36	34
1	QCIF	3.750	30	28
2		7.500		
3		15		
4		30		

Figure 5: The combined scalability of the experiment

In Figure 6, the proposed method is applied to the video sequences including *Bus*, *football* and *Mobile*, where (a) to (c) are the original video, and (d) to (f) are the result of the proposed scheme with Level 1 security (i.e., scrambling the enhancement layers from 17 to 9). Figure 6 (i) to (k) are the results of Level 2 security (i.e., scrambling the enhancement layers from 17 to 1), and Figure 6 (l) to (n) are the results of Level 3 security. The PSNRs of the scrambled video are showed Table 1.

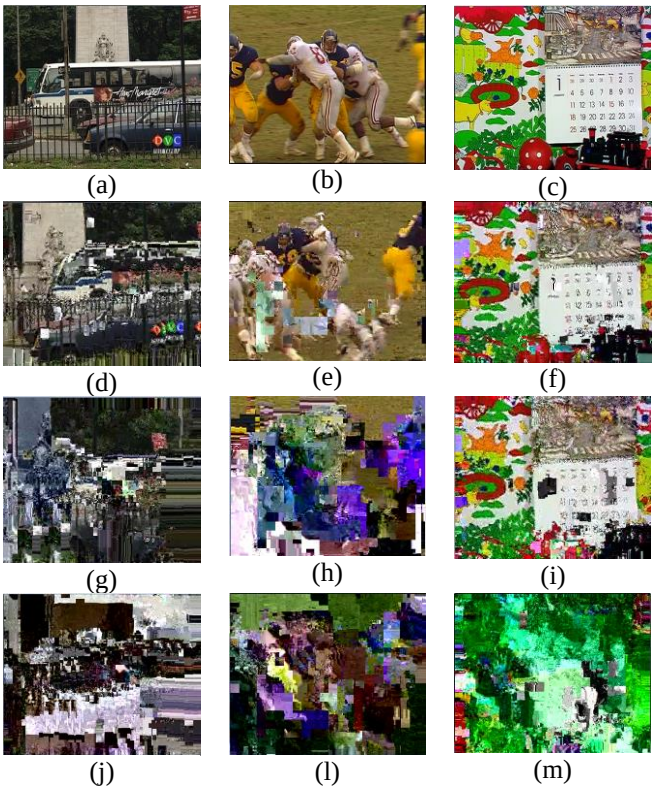


Figure 6: Experimental results of the proposed scheme. (d) to (f) the result of the proposed scheme with scrambling the enhancement layer 17 to enhancement layer 9 at the least frame. (i)-(k) the result of the proposed scheme with scrambling the enhancement layer 17 to enhancement layer 1 at the least frame., and (l)-(n) the result of the proposed scheme with scrambling all layer at the least frame

Table 1: The PSNR values of the original and encrypted video sequences

Video Sequences	PSNR of Encrypted Video
Bus (17 to 9)	15.8226
Football (17 to 9)	19.5327
Mobile (17 to 9)	17.1056
Bus (17 to 1)	13.0513
Football (17 to 1)	13.6889
Mobile (17 to 1)	14.1224
Bus (all)	8.9103
Football (all)	11.0396
Mobile (all)	8.6817

Table 2: The cracking time of the brute force attack in one frame

Encrypted Parameter	Cracking Time
IPM	$8^{16}\times396$
sign bits of non-zero coefficient	$2^{(n\times16)}\times396$,
run_before scrambling	$(n!)^{16}\times396$
sign bits of MVD	$2^{16}\times396$

4.1 Security Analysis

In the security analysis, the cryptographic security refers to the security against cryptographic attacks. The analysis of a brute force attack considers the time of attacking IPM, residual coefficients, and MVD respectively. The time of attacking one frame is shown on **Table 2**. Here, the resolution of the experimental sequence is CIF, in which the total number of the macroblock is 396 in a frame.

1) IPM Encryption

In the process of IPM scrambling, the total amount of encrypted data is $8^{16}\times396$ in one frame, because the possible modes of the coded bitstream are eight for a 4×4 block.

2) Residual Coefficients Encryption

In the process of residual coefficients scrambling, the total amount of scrambled data (i.e., the sign bits of non-zero coefficients) is $2^{(n\times16)}\times396$, where n is the number of the non-zero coefficients. In addition, the total amount of scrambled data of the *run_before* is $(n!)^{16}\times396$, where n is the number of the non-zero coefficients, and the number of possible permutations is $n!$ in a 4×4 block. Hence, the time for breaking *run_before* scrambling is $(n!)^{16}\times396$.

3) MVD Encryption

In the process of MVD scrambling, the total number of scrambled data of sign bits of MVD is $2^{16}\times396$ in one frame under the condition that the inter prediction partition

is always 4×4.

4) Compression Overhead Analysis

Because the encryption is joined to the compression process, the scrambling may impact the compression bit-rate. In the experimental results, there are three different video sequences for testing. **Table 3** shows the experimental results of the sizes between the original video and the scrambled video. From **Table 3** it can observe that the proposed method only slightly increases the bit-rate overhead, since the scrambling process does not increase any non-zero coefficients in the scrambled video.

Table 3: Compression ratio overhead

Video Sequence	Bit-rate Increase(%)
Bus (17 to 9)	0.32
Football (17 to 9)	0.23
Mobile (17 to 9)	0.41
Bus (17 to 1)	0.39
Football (17 to 1)	0.30
Mobile (17 to 1)	0.58
Bus (all)	4.5
Football (all)	4.4
Mobile (all)	4.8

4.2 Complexity Analysis

The computational complexity of the proposed method depends on the size of data to be encrypted. **Table 4** shows the time overhead of the proposed method on the testing video sequence.

Table 4: Time overhead of the proposed scheme

Video Sequence	Time Overhead(%)
Bus (17 to 9)	0.77
Football (17 to 9)	0.93
Mobile (17 to 9)	0.76
Bus (17 to 1)	1.8
Football (17 to 1)	1.5
Mobile (17 to 1)	1.6
Bus (all)	2.6
Football (all)	3.5
Mobile (all)	5.0

1) IPM Scrambling

For the IPM encryption, only the intra prediction modes of 4×4 in each I-frame should be scrambled. Thus, the amount of data that needs to be taken into account in an I-frame is less than (W×H)/(4×4), where W and H represent width and height of the video.

2) Residual Coefficients Scrambling

As to the residual coefficients, the number of sign bits of the non-zero coefficients that needs to be scrambled is about (W×H)/(4×4)×*n*, where *n* is the number of non-zero coefficients. Similarly, the amount of data of *run_before* that needs to be dealt with is about (W×H)/(4×4). Since the scrambling process only applies XOR operations or sign-inversing, the time-consuming overhead can be neglected.

3) MVD Scrambling

As for the MVD, the amount of data that needs to be dealt with is less than (W×H)/(4×4) under the condition that the inter-prediction partition is 4×4.

4.3 Compliance Analysis

In the experimental results, the proposed method still complies with the format of H.264/SVC. Hence, the encrypted video can be played by a codec which is compatible with the H.264/SVC standard.

5 Conclusion

In this paper, a quality controllable video scrambling method is presented to encrypt the significant information and join the encryption process together with H.264/SVC. According to the experimental results and discussions, the proposed encryption scheme provides high security, low computational cost, low bit-rate overhead, and smooth compatibility. Meanwhile, the encrypted H.264/SVC still maintains the scalability property, which means the transcoder can transmit the suitable bit stream to appropriate users on a heterogeneity network. Therefore, the flexibility of the proposed scheme makes it suitable for commerce or entertainment requirements.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant NSC 101-2221-E-415-013.

References

[1] J. Ahn, H. J. Shim, B. Jeon, and I. Choi, "Digital video scramble method using intra prediction mode," in *Proceedings of the 5th Pacific Rim Conference on Advances in Multimedia Information Processing*, pp. 386-393, 2005.

[2] M. Cai, J. Jia, and L. Yan, "An H.264 video encryption algorithm based on entropy coding," in *Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 41-44, 2007.

[3] Y. Fan, J. Wang, T. Ikenage, Y. Tsunoo, and S. Goto, "An unequal secure encryption scheme for H.264/AVC video compression standard," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E91-A, no. 1, pp. 12-21, 2008.

[4] G. M. Hong, C. Yuan, Y. Wang, and Y. Z. Zhong, "A quality-controllable encryption for H.264/AVC video coding," in *Proceedings of Pacific Rim Conference on Multimedia*, pp. 510-517, 2006.

[5] C. Li, C. Yuan, and Y. Zhong, "Layered encryption for scalable video coding," in *Proceedings of 2nd International Congress on Image and Signal Processing*, pp. 1-4, 2009.

[6] S. Li, G. Chen, A. Cheung, B. Bhargave, and K. T. Lo, "On the design of perceptual MPEG-Video encryption

algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 2, pp. 214-223, 2007.

[7] Y. Li, L. Liang, Z. Su, and J. Jiang, "A new video encryption algorithm for H.264," in *Proceedings of Fifth International Conference on Information, Communications and Signal Processing*, pp. 1121-1124, 2005.

[8] S. Lian, J. Sun, D. Zhang, and Z. Wang, "A selective image encryption scheme based on JPEG2000 codec," in *Proceedings of Pacific-Rim Conference on Multimedia*, pp. 65-72, 2004.

[9] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Secure advanced video coding based on selective encryption Algorithm," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, pp. 612-629, 2006.

[10] S. Lian, J. Sun, G. Liu, and Z. Wang, "Efficient video encryption scheme based on advanced video coding," *Multimedia Tool Application*, vol. 38, no. 1, pp. 75-89, 2008.

[11] F. Liu and H. Koenig, "A Survey of video encryption algorithms," *Computers & Security*, vol. 29, no. 1, pp. 3-15, 2010.

[12] E. Magli, M. Grangetto, and G. Olmo, "Transparent encryption techniques for H.264/AVC and H.264/SVC compressed video," *Signal Processing*, vol. 91, no. 5, pp. 1103-1114, 2011.

[13] S. S. Maniccam and N. G. Bourbakis, "Image and video encryption using SCAN patterns," *Pattern Recognition*, vol. 37, no. 1, pp. 725-737, 2004.

[14] S. Park and S. U. Shin, "Efficient selective encryption scheme for the H.264/Scalable video coding(SVC)," in *Proceedings of the Fourth International Conference on Networked Computing and Advanced Information Management*, pp. 371-376, 2008.

[15] M. Pzazrci and V. Dipqin, "A MPEG2-transparent scrambling technique," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 2, pp. 345-355, 2002.

[16] L. Qiao and K. Nahrsedt, "Comparison of MPEG video encryption algorithm," *Computer and Graphics*, vol. 22, no. 4, pp. 437-448, 1998.

[17] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical b pictures," *Joint Video Team, JVT-P014*, no. 1, 2005.

[18] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, 2007.

[19] C. Shi, S. Y. Wang, and B. Bhargave, "MPEG video encryption in real-time using secret key cryptography," in *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, pp. 2822-2828, 1999.

[20] D. Socek, S. Magliveras, D. Culibrk, O. Marques, H. Kalva, and B. Furt, "Digital video encryption algorithms based on correlation-preserving permutations," *EURASIP Journal on Information Security*, vol. 2007, no. 10, pp. 1-8, 2007.

[21] L. Tang, "Methods for encryption and decrypting MPEG video data efficiently," in *Proceedings of the Fourth ACM International Conference on Multimedia*, pp. 219-229, 1996.

[22] C. Wang, H. B. Yu, and M. Zheng, "A DCT-based MPEG-2 transparent scrambling algorithm," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1208-1213, 2003.

[23] X. Wang, N. Zheng, and L. Tian, "Hash key based video encryption scheme for H.264/AVC," *Signal Processing*, vol. 25, no. 6, pp. 427-437, 2010.

[24] Y. Wang, M. Cai, and F. Tang, "Design of a new selective video encryption scheme based on H.264," in *Proceedings of International Conference on Computational Intelligence and Security*, pp. 882-883, 2007.

Ci-Lin Li received his M.S. degree in computer science and information engineering from National Chiayi University in 20011. His research interests include video processing, security, and image processing

Chih-Yang Lin received his Ph.D. degree from Dept. Computer Science and Information Engineering at National Chung-Cheng University, Chiayi, Taiwan in 2006. After graduated, he servered in Advanced Technology Center of Industrial Technology Research Institute of Taiwan (ITRI) from 2007 to 2009. Then, he joined the Institute of Information Science (IIS), Academia Sinica, as a postdoctoral fellow. Currently, he is an Associate Professor in the Department of Computer Science and Information Engineering, Asia University. Prof. Lin has published more than 60 academic papers, and his research interests include computer vision, digital rights management, image processing, and data mining.

Tzung-Her Chen was born in Tainan, Taiwan, in 1967. He received the B.S. degree from the Department of Information and Computer Education, National Taiwan Normal University, Taipei, Taiwan, in 1991, the M.S. degree from the Department of Information Engineering, Feng Chia University, Taichung, Taiwan, in 2001, and the Ph.D. degree from the Department of Computer Science, National Chung Hsing University, Taichung, in 2005. Since August 2008, he has been an Associate Professor with the Department of Computer Science and Information Engineering, National Chiayi University, Chiayi, Taiwan. His current research interests include information hiding, multimedia security, digital rights management, and network security.

International Journal of Network Security, Vol.17, No.3, PP.298-306, May 2015

Improved RSM Algorithm Based on Ensemble Pruning for High-dimensional Steganalysis

Feng-Ying He, Tian-Shun Chen, and Shang-Ping Zhong
(Corresponding author: Feng-Ying He)

College of Mathematics and Computer Science, Fuzhou University
College of Mathematics and Computer Science Fuzhou University, Fuzhou 350108, China
(Email: hfy@fzu.edu.cn)

(Received Apr. 29, 2013; revised and accepted Oct. 1, 2013)

Abstract

Nowadays, there exist a number of challenges in high-dimensional features space for steganalysis, which mainly focus on the difficulty to train classifiers and high computational complexity. Therefore, we propose the improved RSM algorithm based on ensemble pruning for high dimensional steganalysis. Firstly, SFS algorithm is adopted to select the features with high classification ability as fixed features, while the remaining features are selected randomly in the other feature space. Secondly, the feature subset is established using fixed features and the randomly selected features. It is then used to train the base classifiers afterwards. Finally, the pruned ensemble is obtained to yield the final decision using FP-Tree algorithm, in which a FP-Tree is built to compact the prediction results of all base classifiers for the validation set and the ensemble with the best predictive accuracy for the validation set is output. Experiment results demonstrate that the proposed algorithm gains the small-size pruned ensemble and achieves better detection performance with a relatively lower computation overhead when compared with the traditional ensemble pruning algorithms such as Forward Selection and Oriented Order against HUGO steganography.

Keywords: Ensemble pruning, frequent pattern tree (FP-tree), high-dimensional feature, random subspace method, sequential forward selection

1 Introduction

Image blind steganalysis is the method to classify unknown vector by training the extracted feature vectors sensitive to steganography and build models based on the analysis of the impact of the secret information embedded on the digital carrier [13]. The research of [10] has shown that high-dimensional feature vectors for steganalysis can capture more conducive to detect stego information. Increased sophistication of steganographic algorithms together with the desire to detect steganography more accurately prompted steganalysts to use feature vectors with increasing higher dimensionality. For example, [15] constructed a 686-dimensional SPAM feature to detect LSB matching algorithm. Moreover, [9] constructed a 1234-dimensional CDF feature to detect steganographic algorithm YASS, which makes embedding changes in a key-dependent domain. Besides, [6] constructed a 24993-dimensional HOLMES feature proved especially effective against HUGO.

However, the high-dimensional feature space brings heavy burden to the train of classifiers. It will not only increase the calculation complexity, but also may even lead to the dimension disaster [16]. Therefore, how to improve the time efficiency and detection accuracy of the blind steganalysis based on high dimensional feature space has become an urgent subject.

At present, in order to solve the problems caused by high-dimensional feature vectors, scholars at home and abroad mainly use the feature dimensionality reduction techniques, such as principal component analysis (PCA) [17] and independent component analysis (ICA) [3]. However, calculation of the covariance matrix of the original feature vectors is indispensable in the process of dimension reduction, whose time cost is too large for ten thousand dimensional feature vectors, so traditional feature dimensionality reduction technique is not applicable to feature vectors up to thousands of dimensions.

Random Subspace Method (RSM) [8] is a classical ensemble learning algorithm as well as an effective means of dimensionality reduction. The method extracts low-dimensional feature subset from the original high-dimensional feature space, trains based classifiers on each low-dimensional feature subset and combines all the base classifiers' results to present the final decision with the guidance of a certain criteria. The complexity of training the classifiers can be greatly reduced when practicing in low-dimensional space. In addition, RSM just selects a subset of the important features from the original feature space to form a new low-dimensional space and the operation on the global features can be avoided. Therefore, RSM has obvious advantages when processing high-

dimensional features. In 2011, Fridrich emphasis the necessity to use ensemble classification in blind steganalysis and better detection performance is achieved using RSM algorithm to train and classify 24993-dimensional HOLMES features [5, 6, 10].

However, RSM selects features subset randomly, since which the high classification ability of the selected features can not be guaranteed. And it may lead to the poor performance of trained individual classifiers and may even affect integrated performance. Aimed at the deficiency of RSM, many scholars have conducted related research, the major improvements are: 1) combining RSM and other ensemble learning algorithms. [8] improved RSM by combining Boosting and obtained good results with UCI datasets verification, [11], on the other hand, combined RSM and Bagging to improve the classification ability of feature subset. 2）Reducing the randomness of feature subset extraction to improve the classification ability. [18] proposed PCA-based RSM algorithm. PCA was carried out on the sample space and then RSM was adopted to extract feature subspace, as a consequence, it achieved a higher accuracy rate. [19] proposed a semi-random subspace algorithm, global information was added in sub-sample space and experiments have proved its performance improvement in comparison with the traditional RSM algorithm. However, these methods have integrated all the base classifiers, which inevitably leads to a huge computional and storage cost. Besides, with the increase of number of base classifiers, individual differences is more and more difficult to achieve, [21] put forward the concept of "ensemble pruning" and proved better results will be gained by selecting part of base classifiers than using all base classifiers.

Based on the above analysis, this paper proposes an improved RSM algorithm based on ensemble pruning for high dimensional steganalysis. In order to improve the classification ability, the feature subset is no longer randomly selected, instead it selects part of features with better classification ability using Sequential Forward Selection (SFS) as fixed features, while the remaining features are randomly selected from the other feature space. which means the feature subset is made up of fixed features and randomly selected features. Then base classifiers are trained in the feature subset. And by introducing the idea of ensemble pruning, predicted results table are used to store the prediction results of all base classifiers for the validation set and an FP-Tree is built to compact it, finally, the pruned ensemble of classifiers with the best predictive accuracy for the validation set is selected using greedy algorithm to yield classification results. Experiment results show that, the number of classifiers in the ensemble is small and the proposed algorithm achieves the better detection accuracy with lower computational overhead compared with Forward Selection (FS) [2] and Oriented Order (OO) [4] against steganographic algorithm HUGO.

The rest of this paper is organized as follows. In Section 2, we describe the improved RSM algorithm based on ensemble pruning in detail. In Section 3, we implement experiments and present an empirical analysis of the proposed algorithm. Finally, we give the conclusion and some suggestions for future work in the last Section.

2 Improved RSM Based on Ensemble Pruning Algorithm

Ensemble classification is a learning paradigm where a collection of base learners is trained for the same task and then combining their predictions using a certain strategy to get the final predicted results.

Suppose there are *N* base learners, Denote *D* as the expected output of *m* samples, where $D=[d_1, d_2, ..., d_m]^T$, $d_j \in \{-1,+1\}(j=1,2,...,m)$, standing for the expected output of the *jth* sample, f_i (i=1,…,N) as the actual output of the *ith* base learner on *m* samples, where $f_i=[f_{i1}, f_{i2}, ..., f_{im}]^T$, $f_{ij} \in \{-1,+1\}(i=1,2,...,N; j=1,2,...,m)$, standing for the actual output of the *ith* base learner on the *jth* sample. Theory and experiments show that [12], the actual output of the ensemble classifier on the *jth* sample is as follows:

$$\hat{f}_j = \sum_{i=1}^{N} \omega_i f_{ij}, 0 \le \omega_i \le 1 \tag{1}$$

where ω_i is the weight of the base learner, $\sum_{i=1}^{N} \omega_i = 1$.

The generalization error of the *ith* base learner and the ensemble classifier on the *jth* sample *x* is respectively:

$$\varepsilon_{ij}(x) = (f_{ij} - d_j)^2 \tag{2}$$

$$e_j(x) = (\hat{f}_j - d_j)^2 \tag{3}$$

The difference degree of the *ith* base learner relative to the ensemble classifier on the *jth* sample *x* is as follows:

$$a_{ij}(x) = (f_{ij} - \hat{f}_j)^2 \tag{4}$$

Then the generalization error of the ensemble classifier on *m* samples can be expressed as:

$$E = \frac{1}{m}\sum_{j=1}^{m} e_j(x) = \overline{E} - \overline{A} \tag{5}$$

where $\overline{E} = \frac{1}{m}\sum_{i=1}^{N}\sum_{j=1}^{m} \omega_i \varepsilon_{ij}(x)$, standing for the weighted average of the generalization error of all base learners, $\overline{A} = \frac{1}{m}\sum_{i=1}^{N}\sum_{j=1}^{m} \omega_i a_{ij}(x)$, standing for the weighted average of the difference degree of all base learners.

From Formula (5) we can see that increasing the performance of the base learners and the diversity between them helps to improve the generalization ability of the ensemble classifier. But ensemble all base learners inevitably leads to a huge computational and storage cost,

besides, with the increase of the number of base learners, the diversity between base learners is more and more difficult to achieve. [21] proved from the theory analysis and experiment that ensemble some instead of all the available base learners, a better result can be achieved.

Therefore, how to improve the performance of base classifiers, while enhancing the diversity between them and select some base learners with higher diversity and stronger classification ability to make up the ensemble to improve the generalization ability of the ensemble classifier is the main goal of this study.

The main idea of the improved algorithm is that each feature subset is consisted of two parts features to improve the classification ability: One part is the features selected by SFS with better classification ability; the other part is the features selected from the remaining feature space randomly. Then base classifiers are trained on the feature subset, finally, the pruned ensemble E is selected from base classifiers using ensemble pruning algorithm based on FP-Tree to yield the final result. The algorithm flow chart is shown in Figure 1.

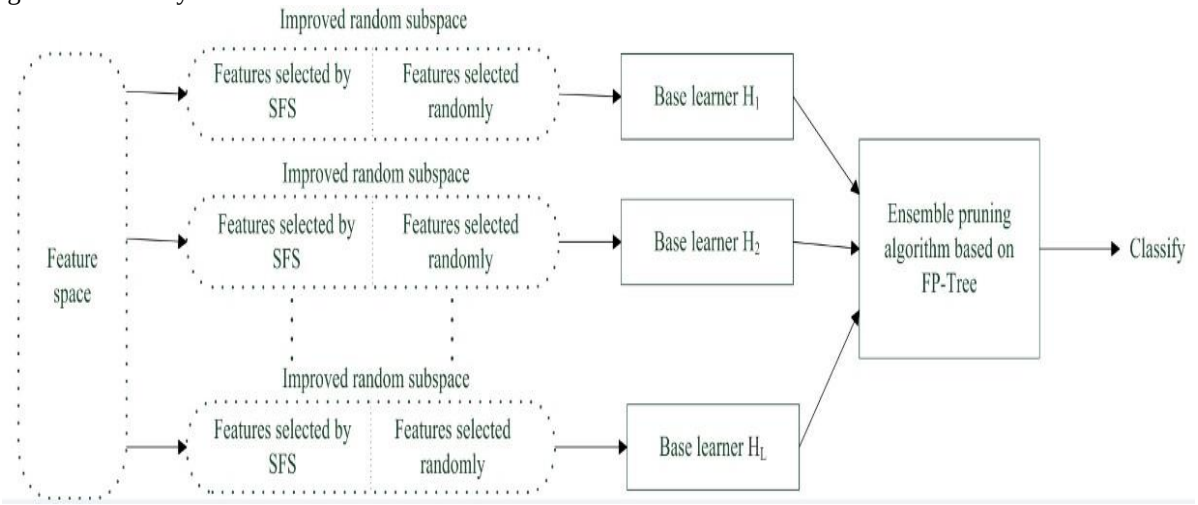


Figure 1: Algorithm flow chart of the proposed algorithm

2.1 Sequence of Forward Selection (SFS) Algorithm

Sequence of forward selection algorithm [14] is a kind of feature selection algorithm. Its essence is a kind of greedy selection algorithm. Assuming feature subset $X = \varnothing$,a feature x_i is selected to join in each loop, which makes the classification accuracy for the validation set V is highest, obtained by classifier trained by classification algorithm C on feature subset X after x_i joining. SFS algorithm is described as follows:

Algorithm 1: SFS Algorithm

1: Begin
2: Initialize the feature subset X as an empty set, $X = \varnothing$.
3: **For** each feature d_i in original feature set D **do**
4: **if** $d_i \notin X$ **then**
5: $X' = X \cup d_i$
6: train classifier using classification algorithm *C* on feature subset X' and calculate classification accuracy T_i for validation set *V*.
7: **end if**
8: **End for**
9: Find out the largest classification accuracy in Step 3 and add the corresponding feature d_i to feature subset X.
10: Count the number of features in X, which is represented with S' if $S' < S$ (S represents the number of selected features), go to Step 3; Otherwise, go to Step 11.
11: Returns the selected feature subset X.
12: End

2.2 Ensemble Pruning Algorithm Based on FP-Tree

To facilitate the description, the ensemble pruning algorithm based on FP-Tree is named CMP-EP algorithm. CMP-EP algorithm converts an ensemble pruning task into a transaction database process, in which the prediction results of all base classifiers for the validation set are organized as a transaction database and an FP-Tree is built to compact it. CMP-EP algorithm selects an ensemble with the best detection accuracy for the validation set as output. Suppose we have obtained L base classifiers by some training methods, now select classifiers to ensemble from *L* base classifiers, the algorithm is described as follows [20]:

Algorithm 2: CMP-EP Algorithm

1: Begin
2: Initialize the pruned classifier *E*, *E.set = NULL*, *E.correct* = 0.
3: Utilize all base classifiers to classify validation set V, remove the classifiers whose prediction accuracy are lower than the threshold f and obtain the prediction table *T*.
4: **For** *k* in 1..*L* **do**
5: Obtain the refining of predictive result table T_k.

6: Utilize T_k to build FP-Tree.
7: select an ensemble of size *k* from Tree using greedy algorithm, if the detection accuracy of the ensemble is higher than that of *E*, then replace *E* with this ensemble.
8: **End for**
9: End

The core of CMP-EP algorithm is described in Step 7, which, in essence, is a greedy algorithm. Its main idea is as follows: Firstly, Path-Table is a two-column table, which stores the classifiers on every path of FP-Tree and the count of correctly-predicted samples. The row with the largest count value in the Path-Table is selected and both of its classifier and count are stored into the variable S. Then the selected low are removed from the Path-Table as well as the related classifiers if exists in the other rows. Afterwards, if there occur rows with the same classifier set, combine them into a new row and set its count value to the sum of count value of the combined rows. Repeat the process of selecting the row with largest count value until the number of selected classifiers equals to k or the Path-Table is empty. More details can be found in [20].

Here we give an example to introduce the CMP-EP algorithm from four aspects: obtain classification results, refine predictive result table, build FP-Tree and select base learners. Denote *L* as the number of base learners, set $L = 8$. The base classifiers are h_1, h_2,…,h_8, respectively. The instances in validation set *V* are identified as X_1, X_2, …, X_{12} in sequence.

1) Obtain classification results

Use *L* base learners to classify the instances in validation set *V* ,the predictive result is stored in the predictive result table, as shown in Table 1. The first column of Table 1 stores the sample, the second column stores the base learners correctly predict the sample and the third column stores the count of base learners correctly predict the sample.

Table 1: Predictive result table

sample	*Base classifier*	*The number of base classifier*
X_1	$h_1,h_2,h_3,h_4,h_5,h_6,h_7,h_8$	8
X_2	h_2,h_3,h_4,h_5,h_7	5
X_3	h_2,h_5,h_6	3
X_4		0
X_5	h_1,h_2,h_6,h_8	4
X_6	$h_1,h_2,h_3,h_4,h_5,h_6,h_7,h_8$	8
X_7	h_5,h_6,h_7	3
X_8	h_2,h_5,h_7	3
X_9	h_3,h_4,h_5,h_7	4
X_{10}	h_1,h_2,h_5,h_6	4
X_{11}	h_2,h_5,h_6	3
X_{12}	h_1,h_2,h_4,h_6,h_7	5

2) Refine predictive result table

According to the principle of majority voting, for an ensemble of size *k*, if the number of classifiers predicting a sample correctly in the ensemble is no less than $\lfloor k/2 \rfloor + 1$, the prediction of the ensemble for the sample must be correct. It is obvious that, if all the classifiers in the ensemble predict a sample correctly, no matter what classifiers are selected, the combined results for the sample are identical. Assuming $k = 5$, the rows whose number values are lower than $\lfloor 5/2 \rfloor + 1$ or equal to 5 should be removed from Table 1, then sort the base classifiers in descend order according to their appearance order. Finally, choose the first $\lfloor 5/2 \rfloor + 1$ classifiers for every row. The refined predictive result table is show in Table 2.

3) Build FP-Tree

The FP-Tree is constructed according to the fourth column of Table 2 (refined base classifier), which is shown as Figure 2. The tree node consists of two parts: the base learner and the count of samples travel the path.

4) Select base learners

The Path-Table with k=5 is constructed from FP-Tree, which is shown as Figure 3(a). The Path-Table is a two-column table, which stores the classifiers on every path of FP-Tree and the count of correctly-predicted samples. According to the greedy algorithm, the row with the largest count value 3 is selected in the Path-Table and both of its classifier and count are stored into the variable *S*, namely *S.set*={h_2,h_5,h_6}, *S.correct*=3. Then we delete the classifiers h_2, h_5, h_6 from Path-Table and combine rows with the same classifier set to one. The updated Path-Table after the first iteration is shown as Figure 3(b). Then the row in the Path-Table with the largest count value 4 is selected, so *S.set*={h_2,h_5,h_6,h_7}, *S.correct*=7. The updated Path-Table after the second iteration is shown as Figure 3(c). In the third iteration, the two rows have the same count value. Considering that the path of h_1 is constructed earlier in Path-Table than that of h_4, we select h_1 to add into *S.set*, *S.set* = {h_2, h_5, h_6, h_7, h_1}. Now, the number of classifiers in *S.set* is equal to 5, so the algorithm is ended and the pruned result *S* is returned, where *S.set* = {h_2, h_5, h_6, h_7, h_1}, and *S.correct* = 8.

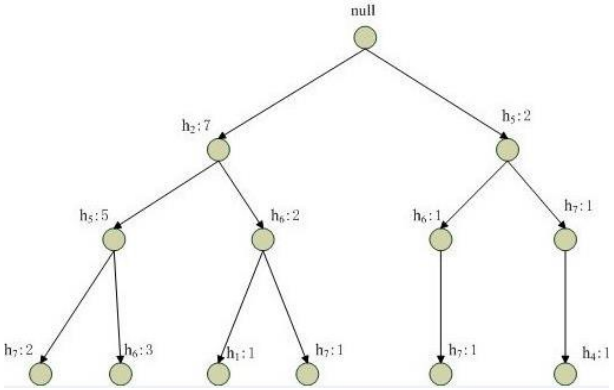


Figure 2: FP-Tree(k=5)

Table 2: Refining of predictive result table

Sample	*base classifier*	*sorted base classifier*	*refined base classifier*
X_2	h_2,h_3,h_4,h_5,h_7	h_2,h_5,h_7,h_4,h_3	h_2,h_5, h_7
X_3	h_2,h_5,h_6	h_2,h_5,h_6	h_2, h_5, h_6
X_5	h_1,h_2,h_6,h_8	h_2,h_6,h_1,h_8	h_2, h_6,h_1
X_7	h_5,h_6,h_7	h_5,h_6,h_7	h_5, h_6, h_7
X_8	h_2,h_5,h_7	h_2,h_5,h_7	h_2, h_5, h_7
X_9	h_3,h_4,h_5,h_7	h_5,h_7,h_4,h_3	h_5, h_7,h_4
X_{10}	h_1,h_2,h_5,h_6	h_2,h_5,h_6,h_1	h_2, h_5, h_6
X_{11}	h_2,h_5,h_6	h_2,h_5,h_6	h_2,h_5, h_6
X_{12}	h_1,h_2,h_4,h_6,h_7	h_2,h_6,h_7,h_1,h_4	h_2, h_6, h_7

2.3 Improved RSM Based on CMP-EP Algorithm

The improved RSM based on CMP-EP algorithm firstly selects the appropriate size of the feature subset r ,then select r /2-dimensional features using SFS algorithm as fixed features, the remaining features are selected randomly in the remaining feature space and finally selects the pruned ensemble E from base classifiers using CMP-EP algorithm to classify samples. The algorithm is described as follows:

Algorithm 3: Improved RSM Based on CMP-EP Algorithm

1: Begin

2: Select *r/2*-dimensional feature subset X with better classification ability using SFS algorithm, $X=SFS(D, r/2, C, V)$, where D represents original feature set, r represents subspace dimension, C represents classification method, V represents validation set and record the position of the feature subset in the original space as *FS*.

3: Select randomly m $r/2$ -dimensional feature subset $R_1, R_2, ..., R_m$ from feature space R ($R=D-X$), and record the position of the corresponding sample subset $S_1, S_2, ..., S_m$, where m represents the number of base classifiers.

4: Construct the feature subset $H_1, H_2, ..., H_m$, where $H_i = X \cup R_i (i = 1, 2, ..., m)$ and obtain base classifiers $C_1, C_2, ..., C_m$ trained on each feature subset with Classification algorithm C .

5: Obtain pruned ensemble E using CMP-EP algorithm.

6: Project FS got in Step 2 onto the testing sample x to obtain the testing sample subset tx .

7: Project $S_1, S_2, ..., S_m$ got in Step 3 on sample subset tr , where $tr = x - tx$ and then obtain random testing sample subset $tr_1, tr_2, ..., tr_m$.

8: Construct testing sample subsets $x_1, x_2, ..., x_m$, where $x_i = tx \cup tr_i (i = 1, 2, ..., m)$.

9: Use pruned ensemble E to classify each x_i $(i = 1, 2, ..., m)$, and get the final decision by majority voting.

10: End

3 Experiment Results and Analysis

3.1 Experimental Setup

To evaluate the performance of the proposed algorithm, we make a comparison among FS algorithm, OO algorithm, FP-RSM algorithm and the proposed algorithm (referred as FP-SFSRSM) from three aspects: Detection accuracy, pruning time and the size of the pruned ensemble.

FS algorithm regards selective ensemble as a stepwise searching process. Each search is based on the evaluation of the previous search. During the experiment, we use the prediction accuracy as the greedy selection criteria.

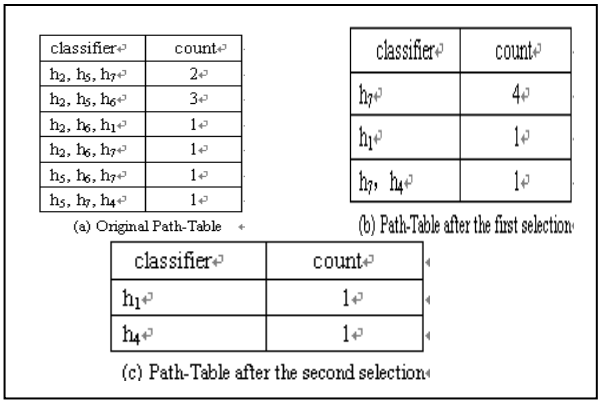


Figure 3: Selecting base classifiers by using Path-Table

OO algorithm sorts all base classifiers using the angle between the base classifier signature vector and the reference vector as evaluation function, then selects base classifiers according to the order. During the experiment, we selects all classifiers whose angle function is less than zero as ensemble classifiers.

FP-RSM algorithm uses the traditional RSM method to extract feature subset while uses the CMP-EP algorithm to select the base classifiers, which is the same as our proposed algorithm.

Experimental data set adopts 10000 images provided by the recent steganalysis competition BOSS [1]. 1000 images are selected randomly as validation set from 10,000 images, 4500 images as a testing set and another 4500 images as a training set, which are embedded by the steganographic algorithm HUGO with payload 0.4bpp. The 12753-dimensional SRM features [7] are extracted for steganalysis. SRM features extract a large number of dependencies of different types between adjacent pixels from a sequence of sub-models. SRM features are sensitive to HUGO thus can effectively against HUGO algorithm.

During the Experiment, we use Fisher Linear Discriminants (FLDs) as base classifiers due to their simple and fast training, the number of base classifiers m is set 51, the classifier threshold f is set 60%, the program code are implemented using C/C^{++} language, the test platform is WIN7 operating system, Intel Xeon E5300 2.60GHz, 8GB

Memory.

3.2 Analysis of Experimental Results

3.2.1 Detection Accuracy

The detection accuracy of the test set is shown in Table 3. To eliminate the randomness, the experiment has been repeated 10 times. The average detection accuracy and its variance are then calculated as the final results. The best results are marked in bold. The average result of different dimension subspace is shown in the last line.

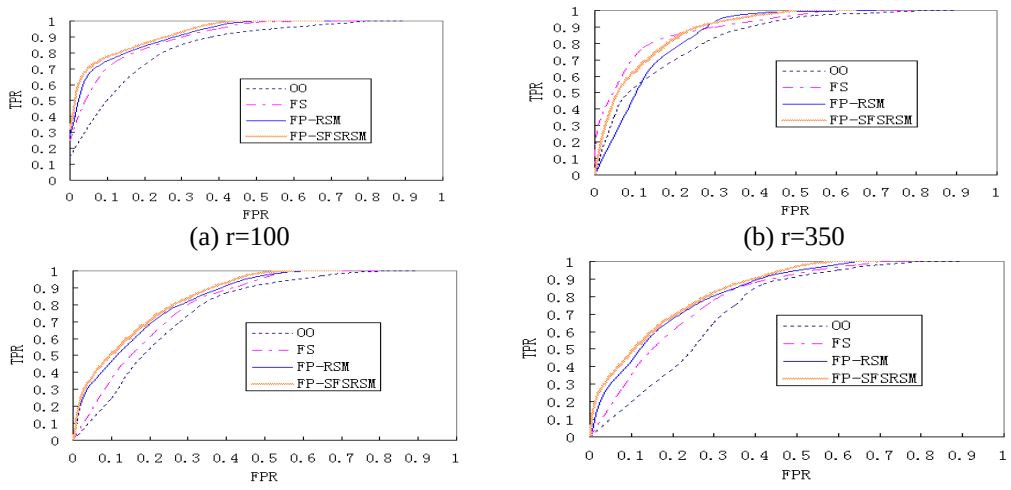
From the results shown in Table 3, it can be seen that:

1) The detection accuracy of FP-SFSRSM algorithm is better than that of FP-RSM algorithm in all the 14 subspaces with different dimensionality, the average detection rate is increased by about 0.4%, which illustrates that the feature selection algorithm of SFS can improve the classification ability of feature subset.
2) FP-SFSRSM and FP-RSM algorithm have obvious advantages in terms of detection accuracy. FP-SFSRSM algorithm gets optimal results in 10 out of 14 subspaces with different dimensionality while FS algorithm gets optimal in the other 4 subspaces. Compared with FS algorithm, FP-RSM algorithm gets better results in 8 subspaces with different dimensionality. The detection accuracy of the OO algorithm is the worst. The main reason is that the chosen process of base classifiers in CMP-EP algorithm is similar to frequent patterns access issues, thus reduce the over-fitting phenomenon to validation set in the selection process, so the accuracy of FP-SFSRSM and FP-RSM algorithm can be improved effectively.

In order to visualize the performance of the algorithms, the Receiver Operating Characteristic (ROC) curve is used when comparing the four kinds of methods. Figure 4 plotted the curves in different dimensions of the subspaces, which are 100,350,600 and 800 respectively. The great advantages of our proposed algorithm in detection accuracy can be obviously seen from the figure.

Table 3: Detection accuracy of four kinds of methods against HUGO

dimension of subspace(r)	*FS* *detection accuracy(%)*	*OO* *detection accuracy (%)*	*FP-RSM* *detection accuracy (%)*	*FP-SFSRSM* *detection accuracy (%)*
100	81.35±1.48	78.75±1.56	82.20±1.12	**82.31±1.07**
150	**82.37±0.62**	78.52±1.83	80.63±1.66	80.78±1.05
200	80.83±1.03	79.18±2.38	82.55±1.85	**83.00±1.64**
250	81.91±2.41	77.62±2.37	82.11±2.22	**82.29±1.45**
300	81.00±3.66	75.57±1.27	80.54±1.39	**81.09±1.15**
350	**82.50±0.87**	76.67±1.89	81.73±1.62	82.35±0.97
400	76.56±0.54	73.32±0.91	77.00±0.80	**77.25±0.63**
450	73.14±2.03	72.13±1.33	74.59±1.43	**75.06±1.39**
500	**81.23±1.12**	75.28±1.21	80.64±1.54	80.91±1.16
600	75.00±1.79	73.10±1.92	75.44±2.09	**75.44±1.20**
700	75.28±2.73	73.55±3.29	75.16±2.25	**75.46±1.93**
800	74.00±1.85	72.64±1.15	74.35±1.72	**74.87±1.11**
900	**76.36±1.32**	72.72±1.39	75.00±1.33	75.28±1.10
1000	72.24±1.10	70.96±1.07	74.62±0.80	**74.93±0.52**
Average	78.13	75.00	78.33	78.64



(a) r=100 (b) r=350

(c) r=600 (d) r=800

Figure 4: ROC curve of four kinds of methods in different dimensions of the subspace

In order to further verify the role of ensemble pruning, the detection accuracy of SFSRSM algorithm and FP-SFSRSM algorithm are compared in different dimension subspaces, as shown in Table 4, where SFSRSM algorithm adopts the method of this paper to extract feature subspace, but integrates all of the base classifiers. TPR represents the true positive rate, TNR represents the true negative rate and AR represents the final detection rate. As can be seen from Table 4, the FP-SFSRSM algorithm increases the performance of the SFSRSM algorithm, gaining about 4% on the average detection rate. Experimental results further proves that selecting a subset of base classifiers can often get better generalization ability than the complete ensemble.

Table 4: Comparison of detection accuracy of SFSRSM and FP-SFSRSM

r	***SFSRSM***			***FP-SFSRSM***		
	TPR	***TNR***	***AR***	***TPR***	***TNR***	***AR***
100	66.55	90.75	78.65	78.31	86.31	82.31
150	92.62	67.78	80.20	86.37	75.19	80.78
200	77.10	83.20	80.15	80.59	85.41	83.00
250	63.15	90.69	76.92	83.38	81.20	82.29
300	66.82	84.72	75.77	80.34	81.84	81.09
350	83.20	73.54	78.37	80.02	84.68	82.35
400	59.63	85.61	72.62	90.55	63.95	77.25
450	71.76	71.90	71.83	66.37	83.75	75.06
500	82.20	72.96	77.58	82.46	79.36	80.91
600	83.72	63.88	73.80	90.69	60.19	75.44
700	59.99	84.35	72.17	69.23	81.69	75.46
800	66.17	77.91	72.04	69.44	80.30	74.87
900	56.82	90.42	73.62	67.81	82.75	75.28
1000	51.30	93.42	72.36	69.56	80.30	74.93

3.2.2 Pruning Time

The pruning time of four kinds of ensemble pruning algorithms are shown in Table 5. The experiments have been repeated for 10 times and the average time is calculated as the final results. Experimental results show that, the pruning time of FP-SFSRSM algorithm and FP-RSM algorithm are close, because they all use CMP-EP algorithm to select classifiers in the ensemble. The pruning time of FP-SFSRSM algorithm and FP-RSM algorithm are better than that of the FS algorithm, but are a bit poor than that of OO algorithm. The main reason is that FS algorithm try to join the remaining base classifiers to ensemble classifier in each loop，which causes a huge computation overhead, but in CMP-EP algorithm, the prediction results of all base classifiers for the validation set are organized as a transaction database, which avoid repeating access to the prediction results, while OO is a ranking-based algorithm, successfully avoid combining predicted results of base classifiers, the pruning time relative to the number of base classifiers is linear, so OO algorithm is fastest.

3.2.3 Size of the Pruned Ensemble

Table 5 also presents the size of obtained ensembles of the four ensemble pruning algorithms, which is derived from the average value of 10 repeating experiments. As can be seen from the results, FP-SFSRSM algorithm selects the least amount of base classifiers, the size of pruned ensemble obtained by FP-RSM algorithm is about 26% of the original base classifiers, similar to that of FP-SFSRSM algorithm. The size of pruned ensemble obtained by FS algorithm is about 35% of the original base classifiers, while the size of pruned ensemble obtained by OO algorithm is the largest, about 51% of the original base classifiers. The ranking method OO algorithm lacks a way to assign an appropriate value to the size of the pruned ensemble, resulting in the largest size of the pruned ensemble. However, according to the average results shown in Table 5, the size of the pruned ensemble obtained by these ensemble pruning algorithms are far less than the original base classifiers, therefore, taking accuracy into consideration, we can get the following conclusion : the ensemble pruning algorithms in experiments can eliminate base classifiers which do not work to improve the generalization ability.

4 Conclusions

The current trend in steganalysis is to train classifiers on feature space with increasing higher dimensionality to obtain more accurate and robust detectors. We present an improved RSM algorithm based on ensemble pruning to solve the problems resulted from high dimensional feature space. In the proposed algorithm, the feature subset is not completely selected randomly, but is composed of two parts: fixed features selected by SFS and features selected randomly, which proved achieve higher accuracy than the classical RSM, and the pruned ensemble is obtained to yield the final results using CMP-EP algorithm, which reduces the over-fitting phenomenon to validation set and avoids repeating access to the prediction results. Experimental results show that, compared with FS algorithm and OO algorithm, the proposed algorithm has a small size of the pruned result and obtains better performance with lower computational overhead against HOGO steganography. The future study will focus on how to appropriately choose the dimension of feature subspace so as to further improve the performance of the algorithm.

Acknowledgments

This study was supported by the Technology Innovation Platform Project of Fujian Province under Grant No.2009J1007, the Natural Science Foundation of Fujian Province under Grant No. 2010J01331. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

Table 5: Pruning time and size of pruned ensembles

r	Pruning time (s)				Size of pruned ensemble			
	FS	OO	FP-RSM	FP-SFSRSM	FS	OO	FP-RSM	FP-SFSRSM
100	8.313	0.052	0.825	0.819	11.10	21.20	15.10	15.00
150	7.914	0.057	0.853	0.853	17.00	19.30	11.20	11.20
200	8.132	0.055	0.772	0.754	15.30	27.00	7.00	6.80
250	9.162	0.059	0.773	0.769	15.20	31.20	7.10	7.00
300	8.832	0.059	0.739	0.711	13.40	31.10	11.40	11.20
350	8.267	0.056	0.703	0.703	23.10	27.00	13.30	13.30
400	7.808	0.057	0.717	0.678	19.70	23.10	11.70	11.30
450	9.312	0.047	0.695	0.680	19.40	25.30	15.40	15.30
500	8.163	0.047	0.814	0.814	23.10	17.70	13.00	13.00
600	7.735	0.062	0.676	0.641	15.50	21.10	19.50	19.20
700	8.624	0.058	0.723	0.721	19.00	23.00	17.20	17.20
800	8.423	0.053	0.683	0.680	13.10	29.10	13.80	13.70
900	7.482	0.060	0.705	0.685	27.10	33.00	13.20	13.00
1000	9.445	0.053	0.663	0.661	23.40	33.00	19.00	19.00
Average	8.935	0.055	0.739	0.726	18.24	25.86	13.42	13.3

References

[1] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system: The ins and outs of organizing BOSS," in *Proceedings of the Information Hiding*, pp. 59-70, Berlin, Germany, Jan. 2011.

[2] R. Caruana, A. Niculescu-Mizil, G. Crew, and et al. " Ensemble selection from libraries of models," in *Proceedings of the Twenty-first International Conference on Machine Learning*, pp. 18, July 2004.

[3] P.Comon, "Independent component analysis," *Higher-Order Statistics*, pp. 29-38,1992.

[4] W. Fan, F. Chu, H. Wang, and et al. "Pruning and dynamic scheduling of cost-sensitive ensembles," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 146-151, London,UK, July 2002.

[5] J. Fridrich, J. Kodovský, V. Holub, and et al. "Breaking HUGO–the process discovery," in *Proceedings of the Information Hiding*, pp. 85-101, Berlin, Germany, Jan. 2011.

[6] J. Fridrich, J. Kodovský, V. Holub, and et al. "Steganalysis of content-adaptive steganography in spatial domain," in *Proceedings of the Information Hiding*, pp. 102-117, Berlin, Germany, Jan. 2011.

[7] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *Information Forensics and Security*, vol. 7, no. 3, pp. 868-882, 2012.

[8] N. Garcıa-Pedrajas and D. Ortiz-Boyer, "Boosting random subspace method," *Neural Networks*, vol. 21, no. 9, pp. 1344-1362, 2008.

[9] J. Kodovský, T. Pevný, and J. Fridrich, "Modern steganalysis can detect YASS," in *Proceedings of the IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics*, pp. 754102-754102-11, California, USA, Feb. 2010.

[10] J. Kodovský and J. Fridrich, "Steganalysis in high dimensions: Fusing classifiers built on random subspaces," in *Proceedings of the IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics*, pp. 78800L-78800L-13, Feb. 2011.

[11] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *Information Forensics and Security*, vol. 7, no. 2, pp. 432-444, 2012.

[12] A. Krogh and J. Vedelsby. "Neural network ensembles, cross validation, and active learning," *Advances in Neural Information Processing Systems*, vol. 8, no. 5, pp. 231-238, 1995.

[13] X. Y. Luo, D. S. Wang, P. Wang, and et al. "A review on blind detection for image steganography," *Signal Processing*, vol. 88, no. 9, pp. 2138-2157, 2008.

[14] K. Z. Mao, "Fast orthogonal forward selection algorithm for feature subset selection," *Neural Networks*, vol. 13, no. 5, pp. 1218-1224, 2002.

[15] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215-224, 2010.

[16] R. L. Somorjai, B. Dolenko, and R. Baumgartner, "Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions," *Bioinformatics*, vol. 19, no. 12, pp. 1484-1491, 2003.

[17] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86,1991.

[18] X. Wang and X. Tang, "Random sampling for subspace face recognition," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 91-104, 2006.

[19] Y. Zhu, J. Liu, and S. Chen, "Semi-random subspace method for face recognition," *Image and Vision Computing*, vol. 27, no. 9, pp. 1358-1370, 2009.

[20] Q. L. Zhao, Y. H. Jiang, M. Xu, and et al. "Fast ensemble pruning algorithm based on FP-Tree," *Journal of Software*, vol. 22, no. 4, pp. 709-721, 2011.

[21] Z. H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1, pp. 239-263, 2002.

Fengying He is the Lecturer in the College of Mathematics and Computer Science, Fuzhou University of China. Her current research interests include machine learning, pattern recognition and multimedia security.

Tianshun Chen is the master in the College of Mathematics and Computer Science, Fuzhou University of China. His current research interests include machine learning and multimedia security.

Shangping Zhong received his PhD in Computer Science and Technology from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a professor with the College of Mathematics and Computer Science, Fuzhou University, China. His current research interests include machine learning, pattern recognition and multimedia security.

On the Security of a Forward-backward Secure Signature Scheme

Liangliang Wang[1], Kefei Chen[2], Xianping Mao[1], and Yongtao Wang[3]
(Corresponding author: Liangliang Wang)

Department of Computer Science and Engineering, Shanghai Jiao Tong University[1]
No. 800 Dongchuan Road, Shanghai 200240, China
School of Science, Hangzhou Normal University[2]
No. 16 Xuelin Street, Hangzhou 310036, China
China Information Technology Security Evaluation Center[3]
No. 8 Yard of Shangdixilu, Beijing 100085, China
(Email: liangliangwang@sjtu.edu.cn)
(Received Apr. 8, 2014; revised and accepted Dec. 24, 2014)

Abstract

In a forward-secure signature scheme, a current exposed secret key can be used to obtain future secret keys which can lead to future signatures invalid. Lin et al. proposed a new method called backward-secure detection to construct forward-backward secure signature schemes to avoid this shortcoming. In this paper, we will show an attack to point out that their scheme doesn't satisfy backward security.

Keywords: Backward-secure detection, cryptanalysis, digital signatures, forward-secure, key exposure

1 Introduction

Nowadays, digital signatures are playing an important role in some electronic applications. However, there is an important problem called key exposure problem exists in traditional digital signatures, that is, once the current secret key is exposed, not only the current signature is not valid, but also the future signatures and the past signatures should not be trusted any more. Secret sharing [6, 7, 8, 15, 18, 19, 20, 22] can reduce the damage caused by the key exposure problem, but distribution of secret pieces is not practical. Forward-secure signatures (FSS) [1, 2, 3, 4, 9, 12, 14], as another partial solution to the key exposure problem, were proposed to provide forward security.

In FSS schemes, the whole lifetime of signature system is divided into discrete time periods. At the end of each time period, the signer obtains a new secret key for the next time period from an old one by a key update algorithm, then the signer deletes the old secret key securely to ensure forward security. Usually, the key update algorithm is a one-way function. Each secret key is only used to sign the message for the corresponding time period. The public key remains unchanged during the whole lifetime. Therefore, in a FSS scheme, even if the current secret key is exposed, signatures from past time periods should still be trusted.

Unfortunately, all forward-secure signatures have a problem that a current exposed secret key can be used to obtain future secret keys which can lead to future signatures invalid. To solve this problem, key-insulated signatures [10, 11, 17, 21] and intrusion-resilient signatures [13] were proposed to achieve a higher level of security. These two new signatures use an additional physical secure device to provide backward security, but this approach is seen as an unpractical approach in many scenarios. In 2001, Mike et al. [5] introduced the concept of strong forward security, but this is not defined in standard forward-secure signatures. Backward-secure detection [16] was proposed to achieve backward security by Lin et al. in 2010. In a forward-backward secure signature (FBSS) scheme, the signer can detect whether a signature is forged after the signature pass the verification.

In this paper, we first review some basic definitions and Lin et al.'s FBSS scheme described in [16]. Then we reanalyze Lin et al.'s FBSS scheme and give a detailed attack to point out that their scheme doesn't satisfy backward security.

The rest of the paper is organized as follows. We briefly review Lin et al.'s FBSS scheme and some preliminaries that will be used in the subsequent sections in Section 2. Then we give an attack to show their scheme doesn't satisfy backward security in Section 3. Finally, Section 4 concludes this paper.

2 Review of Lin et al.'s FBSS Scheme

2.1 Hash Chain

Definition 1 (Hash Chain). *Let $h(\cdot)$ be a collision-resistant one-way hash function, we denote the $h^T(\cdot)$ be a one-way hash chain, where $h^T(\cdot) = h(h^{T-1}(\cdot)) = h(h(h^{T-2}(\cdot))) = \underbrace{(h(h...(h}_{T}(\cdot))...)$ and $h^0(\cdot) = \cdot$. The T is the total number of time periods.*

2.2 Definition

Definition 2 (FBSS). *A FBSS (forward-backward secure signature) consists of the following five probabilistic polynomial-time algorithms:*

Key generation: a probabilistic algorithm, it take as input a security parameter k and the total number of period T, outputs a pair of initial secret key and public key (SK_0, PK).

Signature generation: a probabilistic algorithm, it take as input a message M and a secret key SK_j for the current time period j, outputs a signature σ_j on M for time period j.

Key updating: a deterministic algorithm, it take as input a secret key SK_j for the current time period j, outputs a new secret key SK_{j+1} for the next time period $j+1$.

Signature verification: a deterministic algorithm, it take as input a message M, a public key PK and a candidate signature, outputs a bit b. $b = 1$ means that the signature is generated by using the real secret key, whereas $b = 0$ means invalid signature.

Backward-secure detection: a deterministic algorithm, it take as input a candidate hash value v_{j+1}, outputs a bit c. $c = 1$ means that the signature is accepted, whereas $c = 0$ means rejected.

2.3 Security Requirements

Definition 3. *A forward-secure signature with the properties of backward-secure and backward-secure detection if the verifier receives a signature σ_j after the secret key SK_i has been exposed at time period i for all $j > i$, then the verifier would be able to detect whether the signature has been forge or not, by utilizing the hash chain value $h^{i+1}(x)$.*

2.4 Complexity Assumption

Definition 4 (The Blum Factorization Problem). *Given N, where N is the product of two distinct large primes p and q with roughly the same length $p \equiv q \equiv 3(\mathrm{mod}4)$, find p or q.*

Definition 5 (The Blum Factorization Assumption). *It is computationally infeasible for a probabilistic polynomial-time algorithm to solve the blum factorization problem.*

2.5 Lin et al.'s FBSS Scheme

Lin et al.'s FBSS scheme is proposed by improving Abdalla-Reyzin's FSS scheme [1]. They claim that their scheme can achieve not only forward security but also back-ward security. Their scheme includes five algorithms: Key Generation, Signature Generation, Signature Verification, Backward-Secure Detection and Key Updating. The scheme is constructed as follows.

Key generation: let $p = q = 3 \bmod 4$ be two primes with the same length and $N = pq$ be a k-bit integer. Let H be a collision-resistant one-way hash function, where $H : \{0,1\}^* \rightarrow \{0,1\}^l$. Let Q be the set of non-zero quadratic residues modulo N and $|Q| \geq 2^{k-3}$. The signer randomly select a number $s_0 \in Q$ as the original secret signing key, and compute $u = 1/{s_0}^{2^{l(T+1)}} \bmod N$ as the verification key, where T denotes the total number of time periods and l denotes the length of hash values. The signer publishes the verification key $VK = (N, u, T)$. Finally, the signer randomly select $x \in Z_N^*$ and compute $v_0 = h^T(x)$, here the value x may be seen as the signer's personal secret password.

Signature generation: in order to generate a signature on a message M_j for period j, the signer first randomly selects a number $r_j \in Z_N$ and computes $y_j = {r_j}^{2^{l(T+1-j)}} \bmod N$. Then the signer computes the $j+1$th hash value as $v_{j+1} = h^{T-(j+1)}(x)$. Finally the signer computes $a_j = H(j, v_0, y_j, M_j)$ and $z_j = r_j {s_j}^{a_j} \bmod N$. Thus, the signature for time period j is $\sigma_j = (j, M_j, v_{j+1}, a_j, z_j)$.

Signature verification: on receiving the signature $(j, M_j, v_{j+1}, a_j, z_j)$, the verifier first computes ${y_j}' = {z_j}^{2^{l(T+1-j)}} u^{a_j} \bmod N$, then computes ${a_j}' = H(j, v_0, {y_j}', M_j)$ and checks whether $a_j = {a_j}'$.

Backward-secure detection: after verifying the signature, the verifier can detect whether the signature σ_j is a forged one as follows. The verifier first computes ${v_0}' = h^{j+1}(v_{j+1})$, then checks whether $h^T(x) \equiv v_0 = {v_0}'$.

Key updating: the signer first updates s_{j+1} as $s_{j+1} = s_j^{2^l} \bmod N$, then deletes s_j to guarantee forward security.

3 Cryptanalysis of Lin et al.'s FBSS

3.1 Main Idea

The idea behind our attack begins with the hash value v_{j+1} used in Lin et al.'s FBSS scheme. The hash value v_{j+1} is used in all signatures at time period j and it acts as a partial signature. The attacker who breaks in at time period j maybe perform to forge signatures for time period $j+1$ after it obtains the corresponding hash value v_{j+2}. Therefore, the key problem of the attacker is how to obtains the corresponding hash value which is used to forge. As a partial signature, it is very easy to get it after the signer generates any valid signatures at the corresponding time period. Then, the attacker can use it to forge valid signatures for the corresponding time period. Note that this attack has a restriction, that is, the signer has to sign for the time periods at which the attacker tries to forge signatures before the attacker can forge. Usually in many practical applications of FSS schemes, it is very normal for the signer to generate several signatures at one time period. So in this scenario, the attacker can forge valid signatures for all time periods from which time period it selects to breaks in to the last time period.

3.2 Attack Algorithms

We suppose that the attacker selects time period j to break in and obtains the secret key s_j. Its goal is to construct a new valid signature on a new message M^*_{j+1} for time period $j+1$ to show the scheme doesn't satisfy backward security. After the signer generates a valid signature for time period $j+1$, the attacker can obtain a valid partial signature v_{j+2}. Then the attacker performs as follows.

1) Randomly select $r^*_{j+1} \in Z_N$, then compute $y^*_{j+1} = {r^*_{j+1}}^{2^{l(T+1-(j+1))}} \bmod N$.

2) Use the hash function h to compute $v_0 = h^{j+2}(v_{j+2})$. For a new message M^*_{j+1}, use the hash function H to compute $a^*_{j+1} = H(j+1, v_0, y^*_{j+1}, M^*_{j+1})$.

3) Update s_{j+1} as $s_{j+1} = s_j^{2^l} \bmod N$, then compute $z^*_{j+1} = r^*_{j+1} {s_{j+1}}^{a^*_{j+1}} \bmod N$.

4) The 5-tuple flow $(j+1, M^*_{j+1}, v^*_{j+2}, a^*_{j+1}, z^*_{j+1})$ comprises a new signature on M^*_{j+1} for time period $j+1$ and it can pass the signature verification and the backward-secure detection as as follows.

$$
\begin{aligned}
& {z^*_{j+1}}^{2^{l(T+1-(j+1))}} u^{a^*_{j+1}} \\
= \; & {r^*_{j+1}}^{2^{l(T+1-(j+1))}} {s_{j+1}}^{a^*_{j+1} 2^{l(T+1-(j+1))}} {s_0}^{-2^{l(T+1)} a^*_{j+1}} \\
= \; & {r^*_{j+1}}^{2^{l(T+1-(j+1))}} {s_0}^{2^{l(T+1)} a^*_{j+1}} {s_0}^{-2^{l(T+1)} a^*_{j+1}} \\
= \; & {r^*_{j+1}}^{2^{l(T+1-(j+1))}} \\
= \; & y^*_{j+1}
\end{aligned}
$$

and

$$
\begin{aligned}
h^{j+2}(v^*_{j+2}) &= h^{j+2}(h^{T-(j+1+1)}(x^*)) \\
&= h^T(x^*) \\
&= v^*_0
\end{aligned}
$$

3.3 Another Intuitive Attack Idea

The problem discussed in this subsection is just an intuitive idea, it is necessary to pay enough attention to the problem which we will consider in the following. As described in Lin et al.'s scheme, the verification key of the scheme is (N, u, T). The value v_0 which is obtained from the signer's secret password x is not defined as a part of the verification key. Intuitively, there are not any authentication measures to provide enough trust to the value v_0 which is used to achieve the forward-secure signature with backward-secure detection. Any attacker can replace the value v_0 to attempt to break the backward security. A natural solution to this worry is providing a authentication measure to the value v_0. For example, the signer can acquire a signature of the value v_0 by using the secret signing key. This measure may further improve the FBSS scheme.

4 Conclusions

Forward-secure signatures can reduce the damage caused by the key exposure problem, but once the current secret key is exposed, the future signatures are no longer be trusted. Lin et al.'s new method of backward-secure detection can provide a higher security level to forward-secure signatures. However, their FBSS scheme is insecure against our proposed attack. Moreover, the secret password used in Lin et al.'s scheme may be not the best method to construct FBSS schemes and they did not provide a formal security proof for their scheme. Our future work would be mainly focused on researching on constructing probably-secure FBSS schemes without secret password.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant No. 61133014). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. Abdalla and L. Reyzin, "A new forward-secure digital signature scheme," in *Advances in Cryptology (ASIACRYPT'00)*, pp. 116–129, 2000.

[2] R. Anderson, "Two remarks on public key cryptology," *Unpublished*, 1997. (http://www. cl. cam. ac. uk/users/rja14)

[3] M. Bellare and S. K. Miner, "A forward-secure digital signature scheme," in *Advances in Cryptology (CRYPTO'99)*, pp. 431–448, 1999.

[4] X. Boyen, H. Shacham, E. Shen, and B. Waters, "Forward-secure signatures with untrusted update," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 191–200, 2006.

[5] M. Burmester, V. Chrissikopoulos, P. Kotzanikolaou, and E. Magkos, "Strong forward security," in *Trusted Information*, pp. 109–121, 2001.

[6] T. Y. Chang, M. S. Hwang, and W. P. Yang, "An improvement on the lin-wu (t, n) threshold verifiable multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 163, no. 1, pp. 169–178, 2005.

[7] T. Y. Chang, M. S. Hwang, and W. P. Yang, "A new multi-stage secret sharing scheme using one-way function," *ACM SIGOPS Operating Systems Review*, vol. 39, no. 1, pp. 48–55, 2005.

[8] T. Y. Chang, M. S. Hwang, and W. P. Yang, "An improved multi-stage secret sharing scheme based on the factorization problem," *Information Technology and Control*, vol. 40, no. 3, pp. 246–251, 2011.

[9] S. S. Chow, H. W. Go, C. K. Hui, and S. M. Yiu, "Multiplicative forward-secure threshold signature scheme.," *International Journal Network Security*, vol. 7, no. 3, pp. 397–403, 2008.

[10] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-insulated public key cryptosystems," in *Advances in Cryptology (EUROCRYPT'02)*, pp. 65–82, 2002.

[11] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Strong key-insulated signature schemes," in *Public Key Cryptography (PKC'03)*, pp. 130–144, 2002.

[12] G. Itkis and L. Reyzin, "Forward-secure signatures with optimal signing and verifying," in *Advances in Cryptology (Crypto'01)*, pp. 332–354, 2001.

[13] G. Itkis and L. Reyzin, "Sibir: Signer-base intrusion-resilient signatures," in *Advances in Cryptology (Crypto'02)*, pp. 499–514, 2002.

[14] A. Kozlov and L. Reyzin, "Forward-secure signatures with fast key update," in *Security in Communication Networks*, pp. 241–256, 2003.

[15] C. T. Li and M. S. Hwang, "An online biometrics-based secret sharing scheme for multiparty cryptosystem using smart cards," *Network*, vol. 3, no. 4, p. 5, 2010.

[16] D. R. Lin, C. I. Wang, and D. J. Guan, "A forward-backward secure signature scheme," *Journal of Information Science and Engineering*, vol. 26, no. 6, pp. 2319–2329, 2010.

[17] G. Ohtake, G. Hanaoka, and K. Ogawa, "An Efficient Strong Key-insulated Signature Scheme and Its Application," in *Public Key Infrastructure*, pp. 150–165, 2008.

[18] J. Pieprzyk and X. M. Zhang, "Ideal secret sharing schemes from permutations.," *International Journal Network Security*, vol. 2, no. 3, pp. 238–244, 2006.

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

[20] Y. Tian, C. Peng, and J. Ma, "Publicly verifiable secret sharing schemes using bilinear pairings.," *International Journal Network Security*, vol. 14, no. 3, pp. 142–148, 2012.

[21] J. Weng, X. Li, K. Chen, and S. Liu, "Identity-based parallel key-insulated signature without random oracles.," *Journal Information Science Engineering*, vol. 24, no. 4, pp. 1143–1157, 2008.

[22] C. C. Yang, T. Y. Chang, and M. S. Hwang, "A (t,n) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, 2004.

Liangliang Wang was born in 1984. He is a Ph.D. candidate in Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interests include information security and public key cryptography, etc.

Kefei Chen was born in 1959. He received his Ph.D. degree from Justus Liebig University Giessen, Germany, in 1994. Since 1996, he came to Shanghai Jiao Tong University and became the Professor and the Ph.D. supervisor at Department of Computer Science and Engineering. Since 2013, he came to Hangzhou Normal University and became the Distinguished Professor at School of Science. His research interests include classical and modern cryptography, theory of network security, etc.

Xianping Mao was born in 1982. He is a Ph.D. candidate in Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interests include information security and public key cryptography, etc.

Yongtao Wang was born in 1980. He received his Ph.D. degree in Department of Computer Science and Engineering from Shanghai Jiao Tong University, China, in 2011. Now he is a Research Assistant at China Information Technology Security Evaluation Center, Beijing, China. His research interests include information security and modern cryptography, etc.

An Investigation of the Merkle Signature Scheme for Cryptographically Generated Address Signatures in Mobile IPv6

Sana Qadir, Mohammad Umar Siddiqi and Wajdi Fawzi Mohammed Al-Khateeb
(Corresponding author:Sana Qadir)

Department of Electrical and Computer Engineering, International Islamic University Malaysia
P.O. BOX 10, Kuala Lumpur, 50728, Malaysia
(Email: q57sana@yahoo.com)
(Received Feb. 16, 2014; revised and accepted June 22, 2014)

Abstract

Cryptographically Generated Address (CGA) Signatures are a promising feature of Mobile IPv6 (MIPv6) and are slowly being considered suitable for signing the Binding Update (BU) message. However, the use of the Rivest-Shamir-Adleman (RSA) by the CGA signature algorithms is considered to be a very serious limitation. This is because RSA provides only limited computational security and incurs significant performance overhead. This work investigates the use of an alternative signature scheme called Merkle Signature Scheme (MSS) for use in the CGA signature algorithms. The results show that compared to RSA, MSS provides stronger computational security and significantly improves the performance of the key generation and the CGA signature generation operations.

Keywords: Binding update message, cryptographically generated address signature, Merkle signature scheme

1 Introduction

Security was a specific design consideration for IPv6. The use of existing security mechanisms or protocols like IPsec (Internet Protocol SEcurity) were made compulsory and new features like CGAs and CGA signatures were introduced (see RFC 3972).

MIPv6 provides mobility at the network layer (i.e. IPv6 layer) and it contains several improvements over its predecessor MIPv4. One of the most important improvements is that MIPv6 allows for the direct routing of packets between the Mobile Node (MN) and the Correspondent Node (CN) after the Route Optimization (RO) procedure has been completed (see Figure 1). The Binding Update (BU) message is exchanged during the RO procedure as is shown in Figure 2. The lack of a strong authentication mechanism for the BU message is one of the biggest security vulnerabilities in MIPv6 networks.

The existing authentication mechanism is poor and CGA Signature are considered a very promising alternative [15]. The reason why CGA signatures are considered promising is that they are able to provide strong authentication without requiring nodes to share any common information or security infrastructure [7]. Current key management protocols required by IPsec are not sufficiently scalable or manageable in a global MIPv6 setup [7]. This seriously curtails the use of IPsec and as a result CGA signature-based authentication is gaining wider acceptance. For example, the latest enhancement to MIPv6 called Enhanced Route Optimization (ERO) includes the use of CGA signatures [2]. Although CGA signature-based authentication protocols like ERO are considered reasonably secure, the implementation cost of ERO is not insignificant, especially for low-end MIPv6 nodes like the CN and the MN [14]. In some cases, the cost can be so large that it can be exploited by an adversary to launch a denial of service attack (by flooding the CN with signed BU messages that need to be verified) [14, 15].

The main objective of this work is to show that the performance of CGA signatures can be improved by replacing the use of RSA with the Merkle Signature Scheme (MSS). This should help prevent nodes or networks that use CGA signature based authentication from being vulnerable to denial of service attacks.

2 Related Literature

2.1 CGA Signatures

A node with a CGA can use its private key to sign a packet sent from its address and to assert ownership of its address [3]. A CGA signature is generated using the **CGA signature generation algorithm** (see Figure 3 above) and a CGA signature is verified using a **CGA signature verification algorithm** (see Figure 4 above). Both these algorithms use the RSA Signature Scheme with Appendix

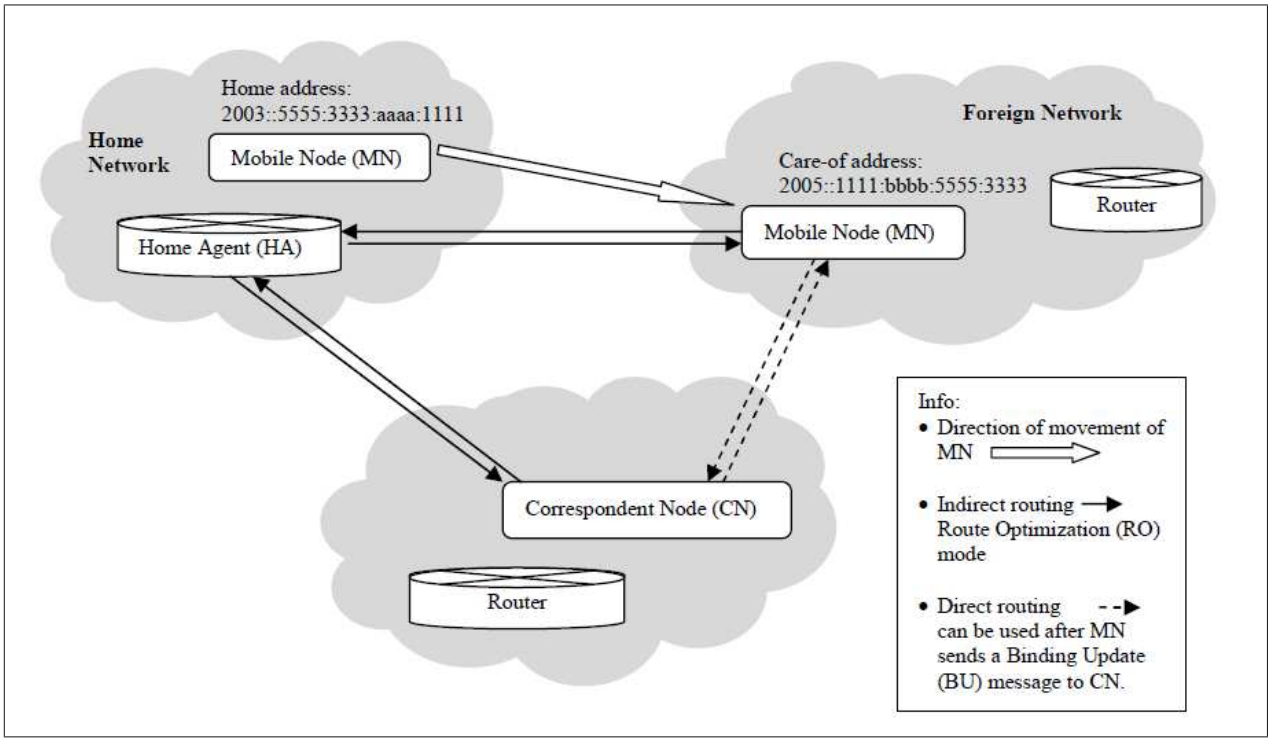


Figure 1: Direct routing in MIPv6

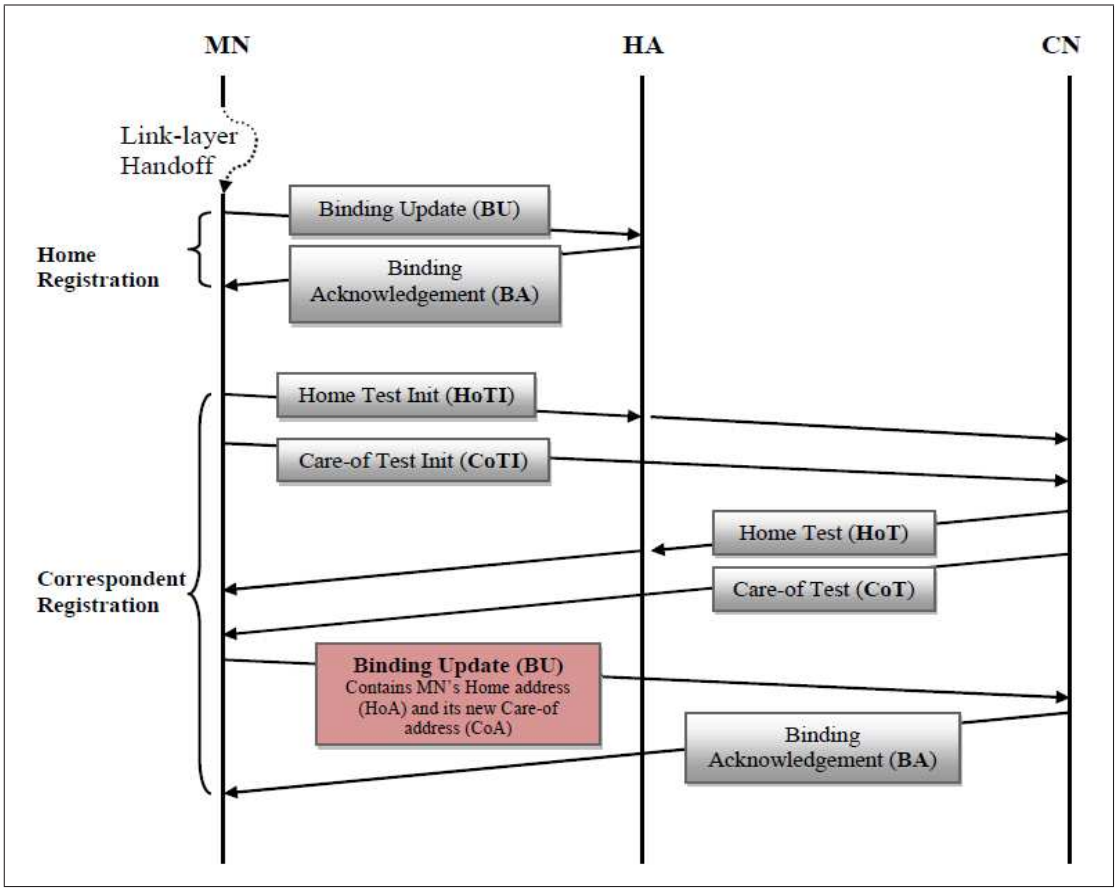


Figure 2: Route optimization procedure [3]

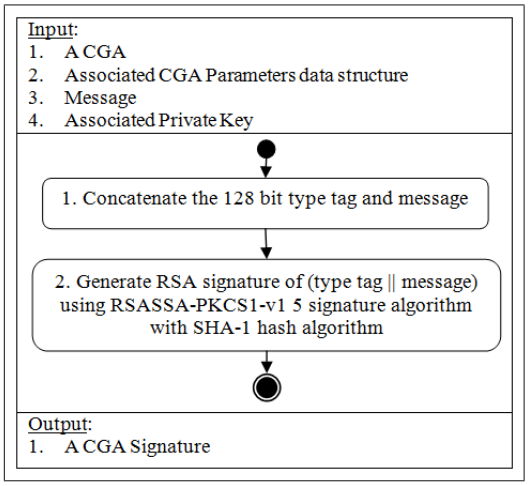


Figure 3: CGA signature generation algorithm [3]

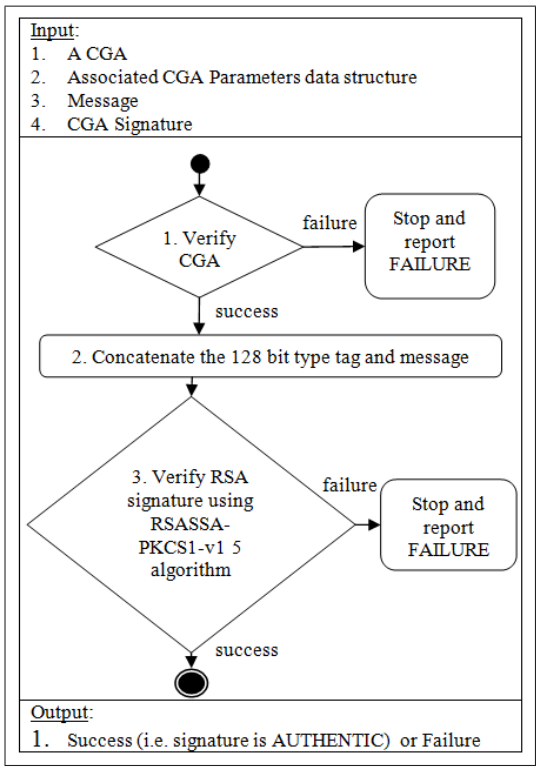


Figure 4: CGA signature verification algorithm [3]

i.e. standardized version 1.5 of Public Key Cryptography Standards #1 (i.e. RSASSA-PKCS1-v1_5). A signature scheme with an appendix means that instead of signing a message directly, a hash function is first used to calculate a digest of the message and then the digest is signed using the private key. This method is popular because the size of the message that can be directly signed must be proportional to the size of the keys and the keys are almost always smaller than the size of the message.

The BU message is sent from a MN to a CN to inform the CN of the MNs new point of attachment [13]. To prevent fake BU messages from being used to launch an attack, a MN must generate a CGA signature and a CN must verify the CGA signature. The use of a CGA signature algorithm to sign a BU message is shown in Figure 5 (on the next page). The sender (i.e. MN) must also share the CGA and the CGA Parameters data structure with the receiver (i.e CN) so that it can verify the CGA signature.

In the SEcure Neighbor Discovery Protocol (SEND), where CGAs were first introduced, the minimum RSA key length required is only 384 bits [3]. RFC 3972 states that if the RSA key is compromised because integer-factoring attacks for the chosen key length have become practical, the key has to be replaced with a longer one [3]. However, existing studies [9, 20] show that:

1) Public key operations are the main contributors to the computational cost of CGA signature generation algorithm;

2) Increasing RSA key lengths significantly degrades the performance of CGA signature algorithms.

Therefore, increasing the RSA key length is neither an acceptable nor a practical solution especially in light of the fact that one of the main reasons for using CGA signatures is to prevent denial-of-service attacks. It is also important that the computational expense on the MN is not large enough to cause significant delays in handover. The most promising solution is to consider replacing the use of RSA with another cryptosystem. Existing studies [8, 9, 19, 20] have investigated alternative signature schemes like Elliptic Curve Digital Signature Algorithm (ECDSA), Digital Signature Algorithm (DSA) and even Modified Feige-Fiat-Shamir (MFFS) scheme. Reference [9] shows that ECC, with its superior performance, small keys and short signatures, is the best alternative to RSA. However, from a performance standpoint, it has one downside: ECDSA signature verification is more costly than RSA signature verification.

2.2 Security of Digital Signatures Schemes

The security of commonly used signature schemes (include ECDSA) depends on the difficulty of solving problems like factoring large composite numbers or computing discrete logarithms. The best cryptanalytic methods of solving these hard problems have sub-exponential run time; meaning that no efficient (i.e. polynomial) algorithm exists to solve these problems. Coupled with the assumption that all adversaries are limited to polynomial time algorithms on classical computers, these schemes are considered secure. However, this assumption is losing credibility in face of the growing acceptance of a near future where quantum computers exist. In 1997, Peter Shor, showed that algorithms for quantum computers (e.g. Shor's algorithm) can solve these hard problems

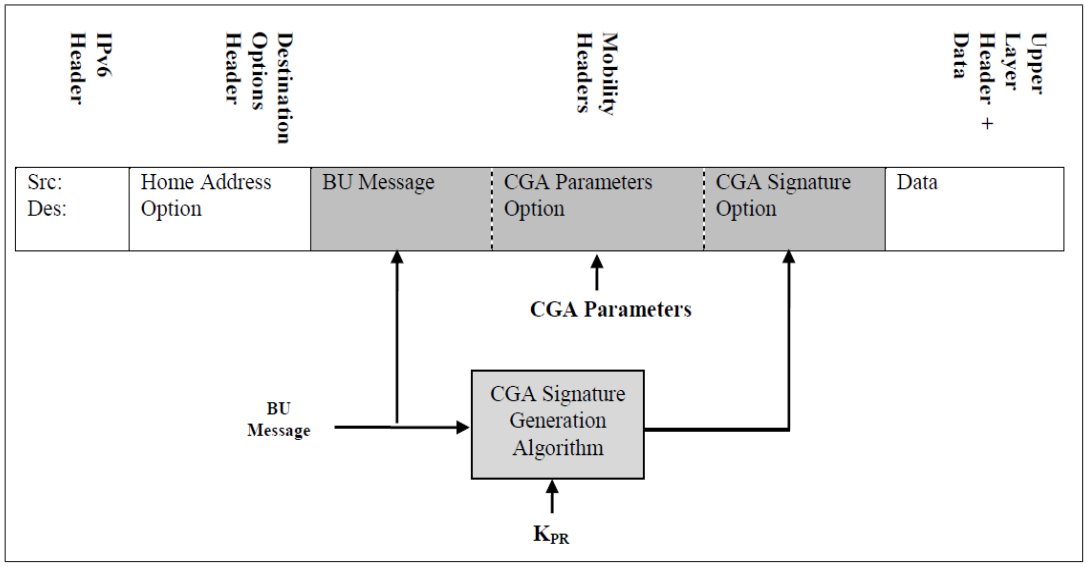


Figure 5: CGA signature of BU message

Table 1: Properties of cryptographic hash functions

Property	Definition	Complexity of Attack Using	
		Classical Computers	Quantum Computers
Pre-image Resistance (One-way)	For any given digest n, it is computationally infeasible to find x such that $g(x) = n$	$O(2^n)$ - Using pigeon hole principle	-
Second-preimage Resistance	For any given block x, it is computationally infeasible to find $y \neq x$ with $g(y) = g(x)$	$O(2^n)$ - Using pigeon hole principle	-
Collision Resistance	It is computationally infeasible to find any pair (x, y) such that $g(x) = g(y)$	$O(2^{n/2})$ - Using birthday paradox	$O(2^{n/3})$

efficiently [4]. As such, it is imperative to consider signature schemes that do not depend on hard mathematical problems.

The signature schemes showing the best potential in a quantum computing scenario are hash-based signature schemes. There are a few other post-quantum cryptosystems that are considered to be extremely difficult to break (e.g. code-based cryptosystems and lattice-based cryptosystems) but hash-based digital signatures schemes have one unparalleled advantage. The security of these schemes depends entirely on the collision resistance property of the hash function used [6]. Refer to Table 1 for a summary of the properties of cryptographic hash functions. If we assume that the hash function used is only vulnerable to generic attacks, then even the fastest quantum algorithm that can be used to attack the collision resistance of a hash function (i.e. Grover's algorithm) has a computational complexity of $O(2^{n/3})$ [6]. So, for example, a hash function with a 256-bit digest will provide computational security of 128-bit security (i.e. $O(2^{128})$) on classical computers and about 85-bit security (i.e. $O(2^{85})$) on quantum computers. Although hash-based digital signature schemes have not formally been proven to resist attacks using quantum computers, their security requirements are minimal and they are independent of hard mathematical problems. In fact, each new hash function can be used to develop a new hash-based signature scheme [6].

2.3 Merkle Signature Scheme

Hash-based signature schemes are based on one-time signature (OTS) schemes and the security of OTS scheme relies on the use of a cryptographically secure hash function. A cryptographically secure hash function is a hash function that is preimage resistant, second preimage resistant and collision resistant (see Table 1).

There are several popular OTS schemes but when selecting a scheme for generating a hash-based signature, it must be kept in mind that the cost of CGA-based au-

thentication depends directly on the performance of the underlying signature scheme. As such, the efficiency and storage requirements of an OTS scheme become the importance selection criteria. In fact, OTS schemes are increasing seen as alternatives to commonly used digital schemes because they are much faster on devices with limited storage and computational resources [5]. This is primarily because they are based on one way functions and one of the most widely accepted class of one-way functions (in cryptography) are hash functions [22].

The most popular OTS scheme, called Lamport-Diffie OTS (LD-OTS) uses large keys and generates huge signatures [4]. In fact, LD-OTS keys and signatures are more than 20 times larger than 1024-bit RSA keys and signatures [4]. The Winternitz OTS (W-OTS) scheme, on the other hand, generates significantly shorter signatures. This scheme provides a parameter, w, through which the signature length can be shortened but at the cost of increased computational cost. The details of W-OTS scheme are given below [6].

Requirements:

1) A one-way function is selected:

$$f : \{0,1\}^n \rightarrow \{0,1\}^n.$$

2) A cryptographic hash function is selected:

$$g : \{0,1\}^* \rightarrow \{0,1\}^n.$$

Key Pair Generation:

1) A Winternitz parameter $w \geq 2$ is selected where w is the number of bits to be signed simultaneously.
2) The t_1, t_2 and t are calculated as follows:

$$\begin{aligned} t_1 &= \lceil \frac{n}{w} \rceil \\ t_2 &= \lceil \frac{\lfloor \log_2 t_1 \rfloor + 1 + w}{w} \rceil \\ t &= t_1 + t_2. \end{aligned}$$

3) The bit strings x_i of the Signature Key X are chosen at random:

$$X = (x_{t-1}, \cdots, x_1, x_0) \in \{0,1\}^{(n,t)}.$$

4) The Verification Key Y is computed:

$$Y = (y_{t-1}, \cdots, y_1, y_0) \in \{0,1\}^{(n,t)},$$

where $y_i = f^{2^w-1}(x_i)$, $0 \leq i \leq t-1$.

W-OTS Signature Generation:

1) Hash digest of the message M is calculated as $g(M) = d$ such that $(d_{n-1}, \cdots, d_0)$.
2) A minimum number of zeros are prepended to d such that the length of d is divisible by w.
3) The extended string d is split into t_1 bit strings $b_{t-1}, \cdots, b_{t-t_1}$ each of length w, i.e. $d = b_{t-1} \,||\, \cdots \,||\, b_{t-t_1}$ where $||$ denotes concatenation.
4) The bit strings b_i are identified with integers in $\{0, 1, \cdots, 2^w - 1\}$ and the checksum c is calculated using:

$$c = \sum_{i=t-t_1}^{t-1} (2^w - b_i).$$

5) A minimum number of zeros are prepended to the binary representation of c such that the length of its binary representation is divisible by w. Then the extended string is split into t_2 blocks: $b_{t_2-1}, \cdots, b_0$ each of length w. Then, $c = b_{t_2-1} \,||\, \cdots \,||\, b_0$.
6) The signature is calculated using:

$$\sigma = (f^{b_{t-1}}(x_{t-1}), \cdots, f^{b_1}(x_1), f^{b_0}(x_0)).$$

W-OTS Signature Verification:

1) For signature $\sigma = (\sigma_{t-1}, \cdots, \sigma_0)$, the bit strings $b_{t-1}, \cdots, b_0$ are calculated as in steps 1 to 5 in W-OTS Signature Generation above.
2) Then, we check if $(f^{2^w-1-b_{t-1}}(\sigma_{n-1}), \cdots, f^{2^w-1-b_0}(\sigma_0)) = (y_{n-1}, \cdots, y_0)$.
3) If the signature is valid, then $\sigma_i = f^{b_i}(x_i)$ and therefore $f^{2^w-1-b_i}(\sigma_i) = f^{2^w-1}(x_i) = y_i$ holds true for $i = t-1, \cdots, 0$.

See Table 2 for an analysis of the computational cost and storage requirements of W-OTS. The flexibility of choosing the value of parameter w and the shorter signatures of W-OTS means that it is considered the best alternative to LD-OTS. Reference [4] states that:

- Signature size of W-OTS varies in inverse proportion to the parameter w.
- Key generation time, signature generation time and signature verification time increase exponentially with the size of parameter w.

An example of the different values for the parameters of W-OTS is given in Table 3. It is evident that, when $w = 32$, the computational cost:

- To calculate the verification key;
- To generate W-OTS signature;
- To verify W-OTS signature.

Total is $t(2^w - 1)$ and therefore infeasible. As such, this work only considers $w = 16$, 8 or 4.

The second point to consider about OTS schemes is that one key pair can be used to sign only one message. This creates an enormous key management problem that renders most OTS schemes more or less

Table 2: Computational cost and storage requirements of W-OTS

	Computational cost	**Storage requirements**
Key generation	$t(2^w - 1)$ evaluations of f	Length of the signature key: $t \times n$ bits Length of verification key: $t \times n$ bits
Signature generation	Worst case: $t(2^w - 1)$ evaluations of f	Length of the signature: $t \times n$ bits
Signature verification	Worst case: $t(2^w - 1)$ evaluations of f	-

Table 3: Different values for parameters of W-OTS

w	n	t_1	t_2	t	$t \times n$	$t(2^w - 1)$
32	256	8	2	10	2560	$4.29e10$
16	256	16	2	18	4608	656350
8	256	32	2	34	8704	2550
4	256	64	3	67	17152	150

impractical. The first step in overcoming this problem was suggested by Merkle and culminated in the creation of the Merkle Signature Scheme (MSS) [17]. MSS requires a cryptographic hash function g and a one-time signature scheme (e.g. W-OTS) [6]. Merkle proposed using a binary hash tree to reduce the validity of an arbitrary but fixed number of one-time verification keys to the validity of one single public key, the root of the hash tree [6]. The details of MSS are outlined below where the height of the hash tree h is 3 [4, 6].

MSS Key Generation (see Figure 6):

1) Select h, such that $h \geq 2$ and $N = 2^h$ (where N is the total number of messages to be signed).
2) Generate 2^h one-time key pairs (X_i, Y_i) where X_i is the OTS signature key and Y_i is the OTS verification key. MSS private key is the sequence of the 2^h OTS signature keys.
3) Calculate the leaves of Merkle tree using $n[0, i] = g(Y_i)$, where $0 \leq i < 2^h$.
4) Calculate the MSS public key (i.e. the root of the Merkle tree). A parent node is the hash value of the concatenation of its left and right children,

$$n[j, i] = g(n[j-1, 2i] \,||\, n[j-1, 2i+1]),$$

where $1 \leq j \leq h, 0 \leq i < 2^{h-j}$.

MSS Signature Generation (see Figure 7):

1) Compute the n-bit digest $d = g(M)$.
2) Generate OTS signature σ_{OTS} of d using the s^{th} OTS signature key X_s.
3) Calculate MSS signature using:

$$\sigma_s = (s, \sigma_{OTS}, Y_s, (a_0, \cdots, a_{h-1})),$$

where:

- s is the index, and
- $(a_0, \ldots, a_{h-1})$ is the authentication path for the verification key Y_s.

MSS Signature Verification:

1) Use the verification key Y_s to verify the one-time signature σ_{OTS} of the digest $d = g(M)$.
2) Validate the authenticity of the one-time verification key Y_s by constructing the path $(p_0, \cdots, p_h)$ from the s^{th} leaf $g(Y_s)$ to the root of the Merkle tree. The verifier uses the index s and the authentication path $(a_0, \cdots, a_{h-1})$ and applies the following construction:

$$p_s = \begin{cases} g(a_{i-1} \,||\, p_{i-1}), & \text{if} \lfloor s/2^{i-1} \rfloor \equiv 1 \bmod 2 \\ g(p_{i-1} \,||\, a_{i-1}), & \text{if} \lfloor s/2^{i-1} \rfloor \equiv 0 \bmod 2 \end{cases}$$

for $i = 1, \cdots, h$ and $p_0 = g(Y_s)$. The authentication of the one-time verification key Y_s is successful if and only if p_h equals the MSS public key.

The Merkle Signature Scheme begins with the signer deciding on the maximum number of messages to be signed with one public key, N. This number of messages must be a power of two and the height of the binary hash tree, h, is calculated as $N = 2^h$. The signer then generates N key pairs (X_i, Y_i) where X_i is the OTS signature key and Y_i is the OTS verification key. The key pairs are used to create a binary hash tree where each leaf is calculated as $n[0, i] = g(Y_i)$

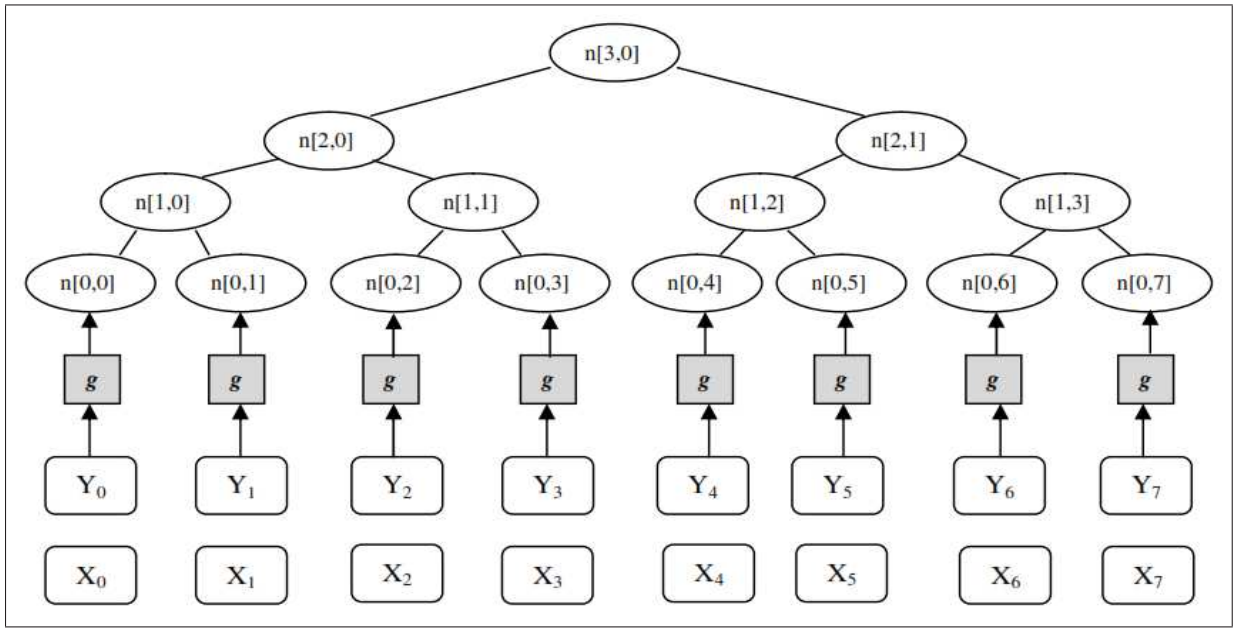


Figure 6: Merkle tree $h = 3$ [4, 6]

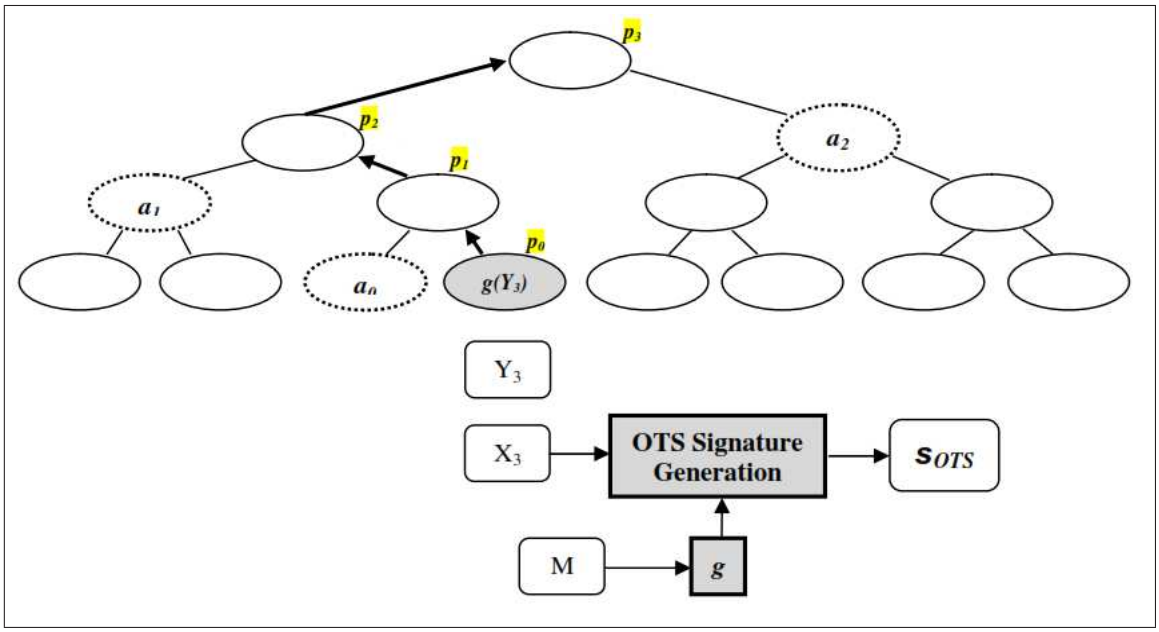


Figure 7: Authentication path for verification key Y_s [4, 6]

and each inner node is the hash value of concatenation of its two children nodes. The root of the hash tree is the MSS public key. The sequence of 2^h OTS signature keys X_i, constitute the MSS private key.

Generating a Merkle signature starts by computing the digest of the message M, $d = g(M)$ and then generating an OTS signature of d using the OTS signature key X_s. Next, the authentication path $(a_o, \cdots, a_{h-1})$ of the OTS verification key Y_s is calculated and sent together with the MSS signature. The verifier begins by verifying the OTS signature using Y_s. If successful, the verifier validates Y_s by calculating the root of the Merkle tree (using Y_s and the authentication path $(a_o, \cdots, a_{h-1})$). If the calculated root of the Merkle tree is equal to the MSS public key sent by the signer, then the signature is considered successfully authenticated.

The strongest advantage of the Merkle signature scheme is its *provable security* - security reductions exist for the Merkle's tree authentication scheme (combined with W-OTS) to the collision resistance of the hash function used [10, 12].

It is also interesting to note that MSS is resistant to differential side channel attacks [4]. Typical techniques for differential side channel attacks require observing the power usage, execution time or electromagnetic signals emitted when different messages are signed using the same private key. This is not possible with MSS because different messages are signed using different private OTS keys.

The only significant drawback of MSS is the relatively expensive generation and storage of the Merkle tree. Table 4 shows an outline of the computation cost and storage requirements of MSS [6].

Generating the entire Merkle tree (in order to calculate the authentication path) for every signature is impracticable especially for big trees. The other alternative of saving all $2^{h+1}1$ nodes results in huge storage requirements. Hence, a good strategy or algorithm is needed to generate a signature in efficient time without having to save too many nodes [4]. This problem is called the *Merkle tree traversal problem*. The traditional algorithm for solving this problem is called the Classical Merkle Tree Traversal [4]. The downside of this algorithm is its storage requirement and existing literature contains proposed improvements to this algorithm as well as some entirely new algorithms to solve the Merkle tree traversal problem (e.g. Logarithmic Merkle Tree Traversal and Fractal / Stratified Tree Traversal) [4, 5, 6]. In the scenario where a CGA-based signature is used to authenticate a BU message (i.e. during route optimization), we do not require a signature scheme capable of signing many messages. In fact, it is better for the security of the network that only about four to eight BU messages can be signed efficiently by the MN before it has to repeat the computationally expensive steps for Key Generation. This is desirable because it will prevent one MN from overwhelming the CN with BU messages that need to be verified - i.e. launching a Denial of Service (DoS) attack.

3 The Design

There are many design choices possible for the underlying hash function g. A hash function that has successfully withstood several years of analysis by the cryptographic community and produces the required digest length can be selected, for example, from the SHA-2 family. This makes MSS a particularly interesting target for implementation research [12]. The hash function can also be chosen in view of the hardware and software resources available. The new hash functions (esp. SHA-3) are investigated in this work. Keccak was selected as winner of the NIST competition for SHA-3 on 2^{nd} October 2012. In spite of the fact that other SHA-3 finalists, like Skein and BLAKE, showed faster performance, Keccak was selected as SHA-3. This was primarily because Keccak is a family of sponge functions and therefore, has a very different design from the SHA-2 family. Also, this study only includes hash functions that produce digests of 256 bits. This is to ensure computational security of at least $O(2^{85})$ on quantum computers and $O(2^{128})$ on classical computers. This is approximately the same as the maximum computational security that is at present achievable in a SEND-enabled MIPv6 network. The reason for this is that generating a CGA on a contemporary CPU using sec value of 1 takes about 402 ms and provides computational security of about $O(2^{75})$ [1, 11]. A CGA that provides higher computational security of $O(2^{91})$ i.e. when sec = 2 will require 1.65 hours to generate. This huge generation time is unacceptable in a MIPv6 environment where address generation must be completed within hundreds of milliseconds [1]. Therefore, all MIPv6 networks at present are limited to using CGAs that have a computational security of about $O(2^{75})$.

4 Implementation

The CGA signature algorithms were coded in C and development was done using the Maemo 5 Software Development Kit (SDK) [16]. The implementation of cryptographic primitives (e.g. RNG and SHA-2) and public-key cryptosystems (i.e. RSA) are from the PolarSSL library [18]. Keccak (i.e. SHA-3), Skein and BLAKE implementations are from the SAPHIR library [21]. The W-OTS scheme was implemented by the authors and then used to implement MSS ($h = 2$). The code was cross-compiled (for ARM architecture) using Scratchbox and run on a Nokia N900. The clock cycles were recorded using the RDTSC instruction. It is important to note that the main processor of a N900 is a 32-bit Cortex A8 running at 600 MHz.

5 Result

Initially, the performance of the different hash functions was recorded for a typical hash operation used in the CGA generation algorithm (i.e. calculation of Hash2) (see Figure 8). It is obvious that SHA-256 provides the best performance in comparison to the other 256-bit hash functions. This performance advantage will guarantee that SHA-256 will continue to be used at least in the near future. BLAKE comes next in terms of performance but it is more than 2.5 times slower than SHA-256. Keccak, although, officially SHA-3, proves to be the slowest hash function included in this study (14 times slower than SHA-256). If, in the future, the SHA-2 hash family needs to be replaced (e.g. because its structure is similar to SHA-1), BLAKE would be the ideal candidate from a performance standpoint.

Figure 9 compares the mean number of clock cycles taken by MSS, RSA-2048 and RSA-4096 for the following operations:

- Key Generation;
- Signature Generation (of BU message);
- Signature Verification (of BU message).

Table 4: Computational and storage requirements of MSS [4]

Operation	**Computational Requirements** (where t and w are parameters of W-OTS; h is the height of the tree and a_i is a node on the authentication path)	**Storage Requirements** (where l is length of digest)
Key Generation	Generate W-OTS key pairs: $2^h \times [t(2^w - 1)$ hash operations + a PRNG operation] Generate MSS public key: $(2^{h+1} - 1)$ hash operations	Entire MSS tree: $(2^{h+1} - 1) \times l$ MSS public key: l
Signature Generation	Generate OTS signature: Worst case: $t(2^w - 1)$ Generate authentication path $a_0, \cdots, a_{h-1}$	$t \times l$ $t \times l + h \times l$
Signature Verification	Verify OTS signature: Worst case: $t(2^w - 1)$ Compute path: h hash operations	

Table 5: Comparison of performance (MSS vs RSA) on a N900 (with a clock rate of 600 MHz)

	RSA-2048		**MSS**		**RSA-4096**	
	Mean no. of clock cycles	Mean time (μs)	Mean no. of clock cycles	Mean time (μs)	Mean no. of clock cycles	Mean time (μs)
Key Generation	12,978,579	21,631	656,134	1,094	138,511,862	230,853
CGA Signature Generation	133,255	222	59,893	100	867,831	1,446
CGA Signature Verification	9,606	16	64,885	108	28,719	48

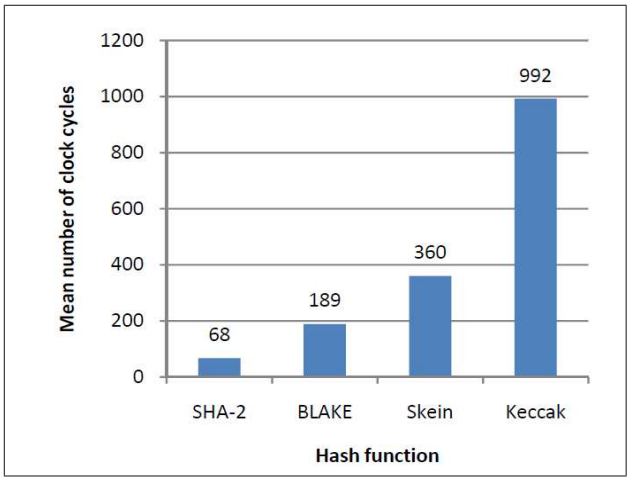


Figure 8: Performance of different hash functions (with 256-bit hash digests)

From the bar charts (Figure 9) and from Table 5, it can be seen that:

- MSS Key Generation time is almost 20 times faster than RSA-2048;
- MSS Signature Generation time is less than half that of RSA-2048;
- MSS Signature Verification time is more than 6 times slower than RSA-2048.

Also, if the performance of MSS is compared to RSA-4096, we find that:

- MSS Key Generation time is 200 times faster,
- MSS Signature Generation time is 14 times faster, but
- MSS Signature Verification time is more than 2 times slower than RSA-4096.

This means that the performance advantage of using MSS improves when compared to increasingly larger RSA key sizes.

A few other points can be deduced from this study:

- When MSS Signatures are used, the computational expense is almost identical for both MN and CN (i.e.

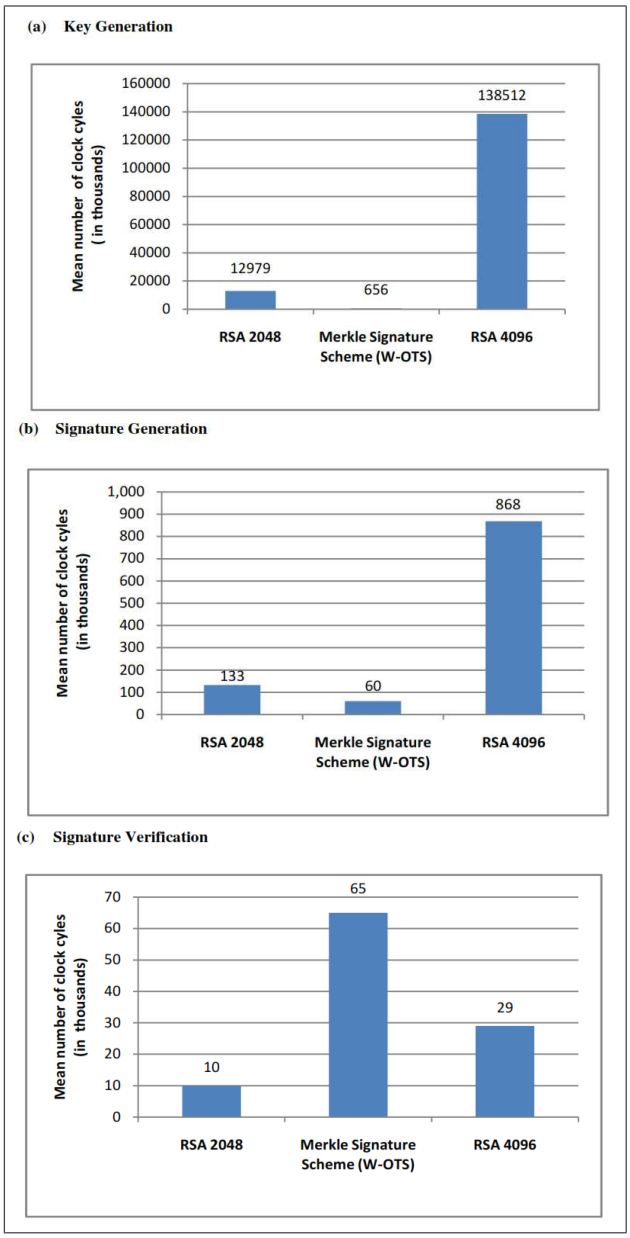


Figure 9: Performance of different hash functions (with 256-bit hash digests)

about 100μs). This is ideal for scenarios where both the MN and CN are low-power mobile nodes.

- MSS public key (32 bytes) is 8 times smaller than RSA-2048 public key (256 bytes).
- MSS signature (2241 bytes) is almost 9 times longer than RSA-2048 (256 bytes) signature.
- The major contributor to computational cost is the generation of the OTS key pairs. It is best therefore to use only four key pairs as in the implementation above ($h = 2$).

6 Conclusion

This work shows that MSS provides an efficient and secure alternative to the use of computationally expensive cryptosystems like RSA for CGA signature. The performance advantage of the MSS Key Generation and Signature Generation Algorithms is very obvious when compared to RSA (esp. for large RSA key sizes). The implementation developed in this study allows for generation of up to four MSS signatures. This means that after four BU messages have been signed, the MN will have to repeat the computationally intensive Key Generation Algorithm. This is an advantage because it will prevent one MN from overwhelming the CN with BU messages that have to be verified (this is one technique of launching a DoS attack). Future work involves bench marking the performance of an actual CGA-based authentication protocol (e.g. ERO) when CGA signatures use MSS with when they use RSA.

References

[1] A. AlSa'deh and C. Meinel, "Secure neighbor discovery: Review, challenges, perspectives, and recommendations," *IEEE Security and Privacy*, vol. 10, no. 4, pp. 26–34, 2012.

[2] J. Arkko, C. Vogt, and W. Haddad, "Enhanced route optimization for mobile IPv6," Tech. Rep. RFC 4866, 2007.

[3] T. Aura, "Cryptographically Gnerated Addresses (CGA)," Tech. Rep. RFC 3972, 2005.

[4] G. Becker, "Merkle signature schemes, merkle trees and their cryptanalysis [Ph.D. Thesis]," 2008. (http://www.emsec.rub.de/media/crypto/attachments/files/2011/04/becker_1.pdf)

[5] D. Berbecaru, "Performance of two one-time signature schemes in space/time constrained environments," in *Proceedings of The 5th IEEE International Symposium of Wireless Pervasive Computing (ISWPC'10)*, pp. 238–243, Modena, Italy, May 2010.

[6] J. Buchmann, E. Dahmen, and M. Szydlo, *Post-quantum cryptography (Eds, Daniel Bernstein, Johannes Buchmann, Erik Dahmen)*, Berlin Heidelberg: Springer-Verlag, 2009.

[7] C. Caicedo, J. Joshi, and S. Tuladhar, "Ipv6 security challenges," *IEEE Computer*, vol. 42, no. 2, pp. 36–42, 2009.
[8] C. Castelluccia, "Cryptographically generated addresses for constrained devices," *Wireless Personal Communications*, vol. 29, pp. 221–232, 2004.
[9] T. Cheneau, A. Boudguiga, and M. Laurent, "Signifiantly improved performances of the cryptographically generated addresses thanks to ECC and GPU," *Computers and Security*, vol. 29, pp. 419–431, 2010.
[10] E. Dahmen and C. Kraub, "Short hash-based signatures for wireless sensor networks," *Lecture Notes in Computer Science*, vol. 5888, pp. 463–476, 2009.
[11] E. Durdagi and A. Buldu, "Ipv4/IPv6 security and threat comparison," in *Procedia Social and Behavioral Sciences (WCES'10)*, vol. 2, pp. 5285–5291, 2010.
[12] T. Eisenbarth, "Cryptography and cryptanalysis for embedded systems [Ph.D. Thesis]," 2009. (http://www.emsec.rub.de/media/crypto/attachments/files/2011/02/thesis_eisenbarth.pdf)
[13] M. S. Hwang, C. C. Lee, and S. K. Chong, "An improved address ownership in mobile IPv6," *Computer Communications*, vol. 31, pp. 3250–3252, 2008.
[14] S. Kuang, R. Elz, and S. Kamolphiwong, "Investigating enhanced route optimization for mobile IPv6," in *Proceedings of The 13th IEEE Asia-Pacific Computer Systems Architecture Conference (ACSAC'08)*, pp. 1–7, Hsinchu, Taiwan, Aug. 2008.
[15] J. Li, P. Zhang, and S. Sampalli, "Improved security mechanism for mobile IPv6," *International Journal of Network Security*, vol. 6, pp. 291–300, 2008.
[16] Maemo.org, "Intro: The home of the maemo community," 2015. (http://maemo.org/intro/)
[17] R. Merkle, "A certified digital signature," 1978. (http://www.cse.msstate.edu/r̃amkumar/merkle1.pdf)
[18] Polarssi, "mbed TLS," 2015. (https://tls.mbed.org/)
[19] S. Qadir, M. U. Siddiqi, and F. Anwar, "Cryptographically Generated Addresses (CGAs): a survey and an analysis of performance for use in mobile environment," *International Journal of Computer Science and Network Security*, vol. 11, no. 2, pp. 24–31, 2011.
[20] S. Qadir, M. U. Siddiqi, and F. Anwar, "A study of cga- (cryptographically generated address) signature based authentication of binding update messages in low-end mIPv6 node," in *Proceedings of The 4th International Conference on Computer and Communication Engineering (ICCCE'12)*, pp. 510–514, Kuala Lumpur, Malaysia, July 2012.
[21] Saphir, "sphlib 3.0," 2015. (http://www.saphir2.com/sphlib/)
[22] M. Stevens, "Attacks on hash functions and applications [phd thesis]," 2012. (http://www.cwi.nl/system/files/PhD-Thesis-Marc-Stevens-Attacks-on-Hash-Functions-and-Applications.pdf)

Sana Qadir received her MSc in Computer and Information Engineering in 2010. She is cuurently a PhD candidate at the Faculty of Engineering in the International Islamic University Malayisia. Her research interests include information security, network security and implementation issues in cryptography.

Mohammad Umar Siddiqi received his B.Sc. and M.Sc. degrees from Aligarh Muslim University (AMU Aligarh) in 1966 and 1971, respectively, and a Ph.D. degree from the Indian Institute of Technology Kanpur (IIT Kanpur) in 1976, all in Electrical Engineering. He has been in the teaching profession throughout, first at AMU Aligarh, then at IIT Kanpur and Multimedia University Malaysia. Currently, he is a Professor in the Faculty of Engineering at International Islamic University Malaysia. His research interests are in coding, cryptography, and information security.

Wajdi F. Al-Khateeb received his MSc. Eng. degree in Telecommunications Engineering from the Technical University Berlin in 1968. After graduation he joined the University of Technology, Baghdad and Northern Petroleum Company, Iraq in 1971 as telecommunications engineer where he assumed various professional engineering activities including senior and chief telecommunications engineer until 1993. In 1995, he joined the Department of Electrical and Computer Engineering, International Islamic University Malaysia. Beside his academic activity, he was appointed as leader of consultancy team to plan, design, and supervise the ICT infrastructure project at the Universitys new campuses in Gombak and Kuantan with more than 30 thousand data/voice nodes to support the ICT applications of the University. He was later conferred a PhD in Engineering from IIUM in 2006. Dr. Wajdi is a professional telecommunications and IT engineer with expert knowledge in telecommunications engineering activities gained through 40 years of experience in many telecommunications systems covering: planning, design, consultation, project management and supervision of wide range of communications systems.

Digital Image Scrambling Algorithm Based on Chaotic Sequence and Decomposition and Recombination of Pixel Values

Dong Wang[1], Chin-Chen Chang[2,3], Yining Liu[4], Guoxiang Song[1], Yunbo Liu[4]
(Corresponding author: Chin-Chen Chang)

School of Mathematics and Statistics, Xidian University, Xian 710071, China[1]
Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan[2]
Department of Computer Science and Information Engineering, Asia university, Taichung 41354, Taiwan[3]
Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China[4]
(Email: alan3c@gmail.com)
(Received Aug. 24, 2014; revised and accepted Jan. 14, 2015)

Abstract

Based on chaotic sequence and decomposition and recombination of pixel values, a new digital image scrambling algorithm is proposed in the paper. While scrambling image pixel values, this new algorithm is able to change the spatial position of pixel, simultaneously scrambling both position and pixel values. Experiments show that the new algorithm is larger in key space, highly efficient, sensitive to secret keys, capable of changing gray-scale feature of images, satisfactory in scrambling result, and resistant to attacks to some extent.

Keywords: Chaotic sequence, decomposition, image scrambling, recombination

1 Introduction

Digital image scrambling technology is an important way of securing digital image information. With the use of transformation techniques, it can change the original image into a disordered one beyond recognition, making it hard for those who get the image in unauthorized manner to extract information of the original image from the scrambled images. Not only can this technology be used for image encryption, but also for digital image watermarking [5, 17] and digital image sharing [14, 15].

Currently, there are numbers of techniques in scrambling digital images [1, 3, 16]. They mainly consist of two categories. One is to scramble pixel spatial position of images, represented by Arnold transformation, baker transformation, magic transformation, Hilbert curve,Gray code transformation, etc. [4, 6, 7, 8, 10, 11]. Arnold matrix transformation is the most typical one and many scholars have made lots of researches into Arnold matrix and the scrambling method: brief introduction to digital image scrambling technology based on Arnold transformation was made in the literature [13], where methods on scrambling digital images in positional space and color space were involved; the literature [9] made improvements on techniques of scrambling digital images with Arnold transformation, and security of the algorithm was reinforced by the introduction of secret keys into scrambling algorithm. The other is to scramble pixel values of digital images by pseudo-random number.

Many in-depth researches into this category have also been made: in the literature [2], digital images were scrambled and restored with the use of random upper (lower) triangular reversible matrix, and this method is of great application value due to its easy operation in encryption and decryption; based on Arnold transformation, the literature [19] proposed new technologies of using matrix transformation, under the control of secret keys, to scramble and restore digital images,and achieved satisfactory results in encryption and decryption; the literature [18], based on the image scrambling concept of gray-scale transformation, scrambled image pixels by using Exclusive-OR operation, and its algorithm features high execution efficiency; the literature [12] made use of the characteristics of chaotic sequence, namely easy to generate and sensitive to initial conditions, and proposed image encryption algorithm based on chaotic sequence, which achieved desirable encryption effect.

Based on Logistic chaotic sequence, the paper designed a new image scrambling algorithm which first decomposes pixels value of digital images, and then recombines the decomposed pixels by Logistic chaotic sequence. Able to synchronically change the pixel value and spatial position of images and diffuse errors, this algorithm is relatively secure.

2 Description of Image Scrambling Algorithm

2.1 Logistic Mapping, Decomposition and Recombination of Pixel Values

Logistic mapping is a kind of simple but widely-used dynamic system, and its mathematical expression is shown as follows:

$$x_{k+1} = f(x_k) = \mu x_k(1 - x_k)$$

where μ is a constant, $x_k \epsilon (0, 1)$, $k \epsilon N$. When $3.569945 < \mu \leq 4$, this mapping comes into chaos state. In other words, sequence generated by this mapping is characterized by certainty, pseudo-randomness, aperiodicity, non-convergence and sensitiveness to initial value.

An image can be defined as a two-dimensional function $f(x, y)$, of which x and y stand for spatial coordinates. f is the gray value of the image at arbitrary point (x, y). The gray level represented on computer has 256 scales,ranged from 0 to 255. We can decompose and recombine pixel value of digital images. The pixel value is decomposed in the order of hundreds place, tens place and units digit. Then, a new pixel is recombined by randomly selecting one pixel from the group of hundreds place, tens place and units digit, respectively. Not only is the newly-recombined pixel different from the original one in gray value, but also its hundreds place, tens place and units digit come from other different pixels. As a result, the pixel spatial position is scrambled as well. Therefore,digital images scrambled in this way become disordered in both pixel position and pixel gray value, indicating it can scramble pixel position and gray value simultaneously. However, when pixel is decomposed, the group of hundreds place $\{0, 1, 2\}$, and the other two groups, in tens place and units digit, are of the same, $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. When we randomly select elements from the group of hundreds place, tens place and units digit, it's likely that the value of the newly-combined element is more than 255. For example, the two pixels, respectively 205 and 189 in pixel values, are likely to be transformed into two different pixels with respective pixel values of 285 and 109. However, the pixel with pixel value of 285 goes beyond the scope of image gray scale represented on computers.

To solve this problem,the pixel value is firstly converted from decimalism into quaternary. The maximum number that a four-digit quaternary number can represent is 3333_4, which comes exactly to 255 when converted into decimalism. Therefore, if the decimalism is converted into quaternary at first, however the pixel is decomposed and recombined, the pixel value will always remain within the scope of image gray scale represented on computers.

2.2 Principles and Steps of the Algorithm

The first step is to read in a digital image and extract gray matrix p, of the digital image. The gray matrix P is then straightened into a vector $p(t)$, of which t stands for the number of image pixel. Each pixel is converted into a quaternary, and the first digit from left to right is put into a vector. The rest can be done in the same manner. Thus, four vectors are generated, namely $r_1(t)$, $r_2(t)$, $r_3(t)$ and $r_4(t)$.

The real numbers of μ_1 and x_0^1 are selected as secret keys, and a double precision chaotic sequence $\{x_1, x_2, x_3, \cdots, x_t\}$ is generated with the use of Logistics mapping. The numbers in chaotic sequence are arranged in ascending order and a new sequence, $\{x_1^{'}, x_2^{'}, x_3^{'}, \cdots, x_t^{'}\}$ is produced. The location code j of each element x_t of chaotic sequence $\{x_1^{'}, x_2^{'}, x_3^{'}, \cdots, x_t^{'}\}$ in the new sequence is then determined. Thus, a binary collection, $\{(i, j)|1 \leq i \leq t, 1 \leq j \leq t\}$ is produced. The binary collection acts on $r_1(t)$. In other words the i^{th} element is placed at the j^{th} position to generate a new vector $r_1^{'}(t)$. The above process is repeated. $\{\mu_2, x_0^2\}$, $\{\mu_3, x_0^3\}$, $\{\mu_4, x_0^4\}$, reselected to act on the three remaining vectors in turn, and $r_2^{'}(t)$, $r_3^{'}(t)$, $r_4^{'}(t)$ are respectively generated.

From left to right, element of $r_1^{'}(t)$ is regarded as the first digit, and $r_2^{'}(t)$, $r_3^{'}(t)$, $r_4^{'}(t)$ as the second, third and fourth digit. The quaternary is converted into decimalism and a new pixel matrix $P^{'}$ is generated.Finally, a scrambled image is obtained.

Specific steps for scrambling encryption are listed as follows:

1) Read in information of original images and input encryption keys x_0, μ, of which $x_0^i \epsilon (0, 1)$, $\mu_i \epsilon (3.569945, 4)$, $i = 1, 2, 3, 4$.

2) Convert the pixel value from decimalism into quaternary, and decompose them into four parts.

3) Produce Logistic chaotic sequence control according to x_0, μ and recombine the new pixels.

4) Convert the recombined pixel from quaternary into decimalism.

5) Encryption is done and show the scrambling image.

As decryption is the inverse process of encryption, the original images can be restored when correct secret keys are input and reverse operation is done on encryption.

3 Experimental Results and Performance Analysis

Simulation experiment was complied and operated in Matlab2010. Hardware configuration was: Pentium(R) Dual-Core 3.0G, 2G RAM. Standard Lena grey-scale map, 512×512, was used for the test. Practical secret keys (μ_0, x_0) were: (0.56, 3.71), (0.78, 3.63), (0.27, 3.81) and (0.63, 3.91). The following results were obtained from the test:

Several comments on the simulation results are as follows.

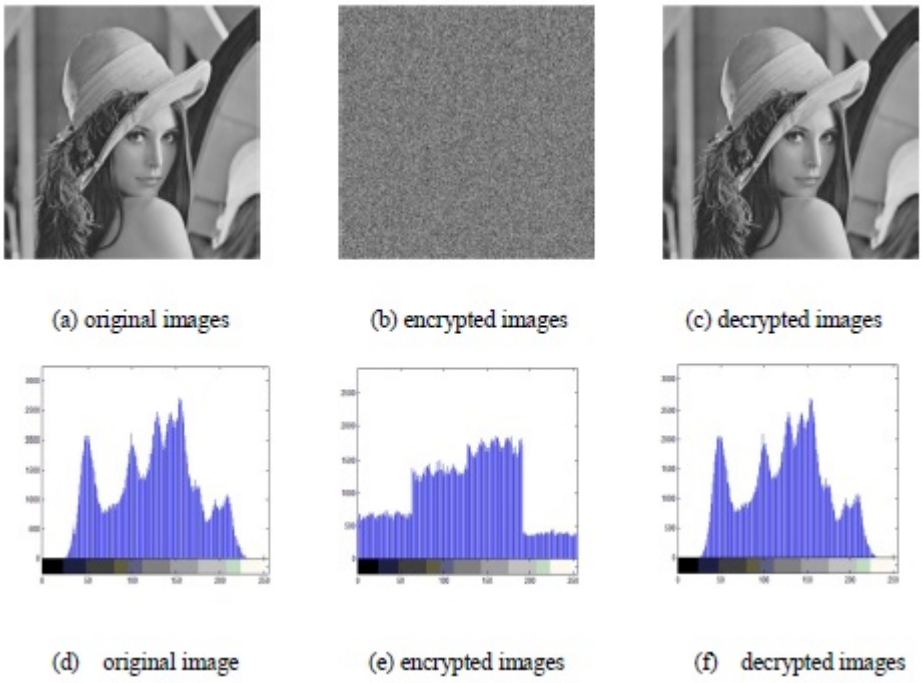


Figure 1: Images and their grayscale distribution before and after scrambling

1) Algorithm in this paper depends only on chaotic sequence and its sorting, so it can be implemented easily. From encrypted and decrypted images and gray histogram of encrypted and decrypted images in Figure 1, we can see that this algorithm was able to change the spatial position and gray-scale features of pixels at the same time. In addition, the conversion in positional notation avoided the overflow of gray value during pixel value recombination. In terms of effect, images scrambled with this method were fine in texture and uniform in diameter. In terms of human visual effect, encrypted images were completely disordered and no image outline could be traced, indicating it was hard to detect information of the original images.

2) As logistic chaotic sequence is extremely sensitive to initial value, this algorithm is quite sensitive to secret keys. Even the slightest perturbations to secret keys will lead to failure in restoring image information. The secret keys used for encryption in Figure 2 were respectively $(0.56, 3.71)$, $(0.78, 3.63)$, $(0.27, 3.81)$ and $(0.63, 3.91)$. The secret keys used for unauthorized decryption I were respectively (0.5600000000001, 3.71), $(0.78, 3.63)$, $(0.27, 3.81)$ and (0.6300000001, 3.91); $(0.56, 3.71)$, (0.78, 3.6300000000001), (0.27, 3.8100000000001) and $(0.63, 3.91)$ for unauthorized decryption II; (0.5600000001, 3.710000000001), (0.78, 3.6300000001), (0.2700000001, 3.81) and $(0.63, 3.91)$ for unauthorized decryption III.We can see from the figure that small perturbations to secret keys led to complete failure in restoring the images.

3) Figure 3 shows that the restored images produced many noises when the scrambled image information was modified in unauthorized manner or concealed and the pixel values were changed. (e) decryption image when 900 pixels are attacked (f) decryption image when $64,000$ pixels are attacked (g) decryption image when $\frac{1}{4}$ pixels are attacked. What's more, the noise might spread throughout the whole image. Therefore, this algorithm features strong shear resistance.

4) Three channels of color image RGB can be respectively seen as three grayscale sequences and this algorithm is applied to scramble them respectively. This algorithm can be used to encrypt color images. Besides, encryption algorithm in this paper is to straighten the image matrix and transform it into vectors. Therefore, it is feasible to encrypt and decrypt images of any size. Figure 4 shows the result of encryption and decryption when the algorithm, discussed in the paper, was applied to 24-scale true color image of 640×480.

4 Conclusions

The proposed digital image scrambling algorithm based on chaotic sequence and decomposition and recombination of pixel values is able to simultaneously scramble pixel positions and pixel values of images. Through decomposition and recombination of pixels, the algorithm scrambles pixel positions and changes pixel values. During recombination, inflow of pixel values is avoided by

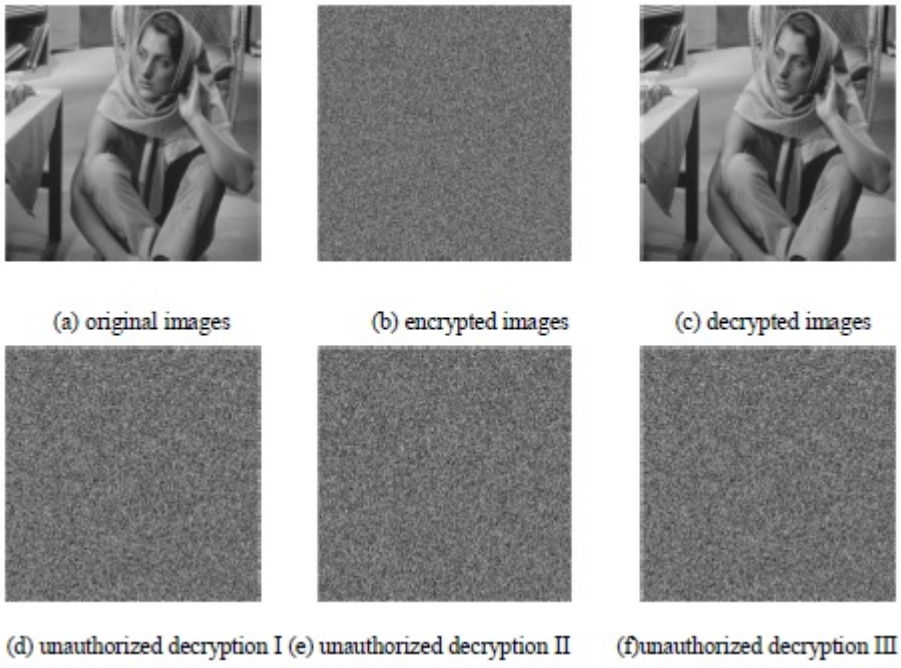


Figure 2: Influence of small perturbations of secret keys on decryption

conversion of number systems. Apart from disordering pixel positions and changing pixel values, this algorithm is able to diffuse errors, i.e. it is capable of spreading the errors in a particular area to the whole image in the form of noise. From the experimental results, we see that our method is indeed resistant to attacks and relatively safe.

Acknowledgments

The work described in this paper was supported by the National Science Foundation of China (Grant No. 61363069, 61362021, 11201094, 11101100), Guangxi Natura Science Foundation (Grant No. 2012GXNSFBA053014, 2013GXNSFDA019030, 2012GXNSFBA053006, 2014GXNSFAA118364), and the High Level Innovation Team of Guangxi Colleges and Universities, and Program for Innovative Research Team of Guilin University of Electronic Technology.

References

[1] M. Amin, O. S. Faragallah, A. A. Abd El-Latif, "A chaotic block cipher algorithm for image cryptosystems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3484-3497, 2010.

[2] Y. Chen, S. Zhang, "A novel digital image scrambling method based on a class of stochastic matrices", *Journal of Southern Yangtze University (Natural Science Edition)*, vol. 5, no. 1, pp. 6-9, 2006.

[3] A. A. Abd El-Latif, L. Li, N. Wang, X. Niu, "Image encryption scheme of pixel bit based on combination of chaotic systems," in *Proceedings of 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'11)*, pp. 369-373, 2011.

[4] F. Han, J. Hu, et al., "A biometric encryption approach incorporating fingerprint indexing in key generation", in *Proceedings of International Conference on Biometrics*, pp. 675-681, Kunming, China, 2006.

[5] M. S. Hwang, K. F. Hwang, C. C. Chang, "A time-stamping protocol for digital watermarking," *Applied Mathematics and Computation*, vol. 169, pp. 1276-1284, 2005.

[6] B. Li, J. W. Xu, "Period of arnold transformation and its application in image scrambling", *Journal of Central South University of Technology*, vol. 12, no. 1, pp. 278-282, 2005.

[7] X. Lin, L. Cai, "Scrambling research of digital image based on Hilbert curve", *Chinese Journal of Stereology and Image Analysis*, vol. 9, no. 4, pp. 224-227, 2004.

[8] J. Shen, X. Jin, C. Zhou, "A color image encryption algorithm based on magic cube transformation and modular arithmetic operation", in *Proceedings of the 6th Pacific Rim Conference on Multimedia*, pp. 270-280, Jeju Island, Korea, 2005.

[9] Y. Si, B. Kang, "Digital image scrambling based on improved arnold transformation", *Computer Technology and Development*, vol. 18, no. 2, pp. 74-77, 2008.

[10] Z. Tang, X. Zhang, "Secure image encryption without size limitation using arnold transform and ran-

][!ht]

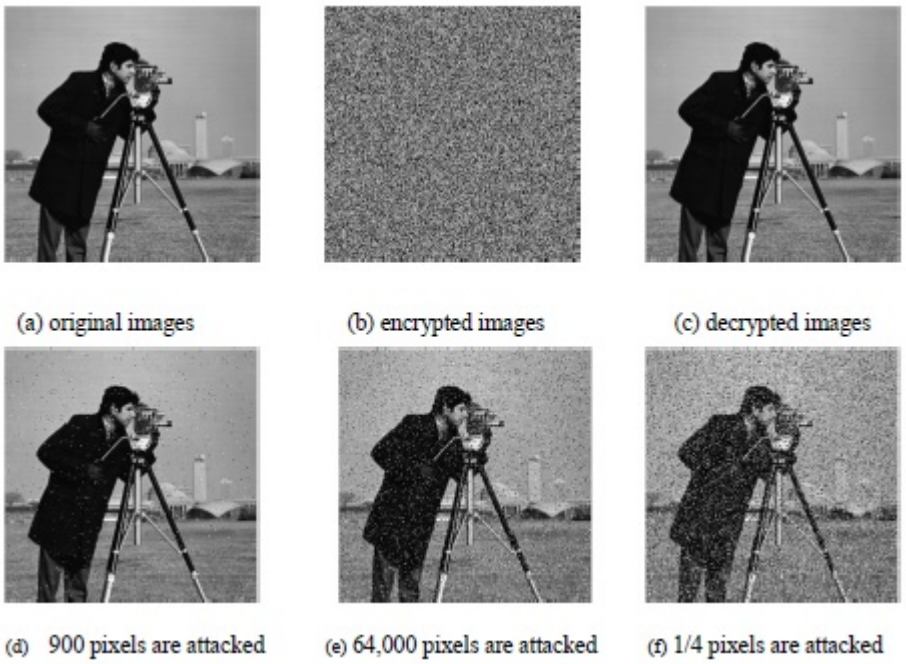


Figure 3: Test effect on pixel attacked

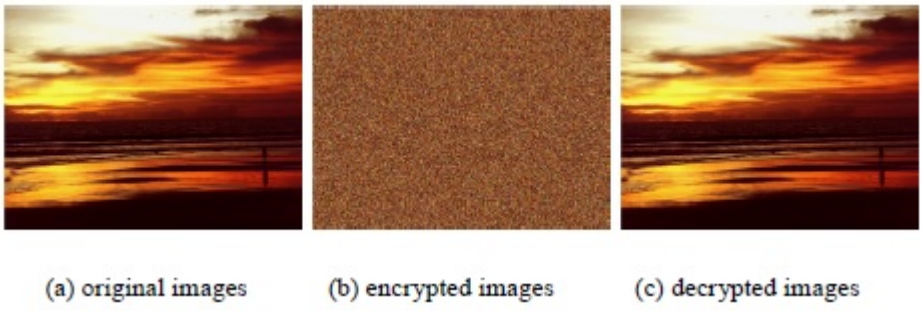


Figure 4: Encryption and decryption of color images

dom strategies", *Journal of Multimedia*, vol. 6, no. 2, pp. 202-206, 2011.

[11] D. Wang, Y. Jin, "Semi-period of doubly even order magic square transformed digital image", *Journal of Zhejiang University (Science Edition)*, vol. 32, no. 3, pp. 273-276, 2005.

[12] Y. Wang, et al., "A new chaos-based fast image encryption algorithm", *Applied Soft Computing*, vol. 11, pp. 514-522, 2011.

[13] D. Wei, et al., "Digital image scrambling technology based on Arnold transformation", *Journal of Computer-aided Design and Computer Graphics*, vol. 13, no. 4, pp. 338-341, 2001.

[14] C. C. Wu, M. S. Hwang, and S. J. Kao, "A new approach to the secret image sharing with steganography and authentication," *Imaging Science Journal*, vol. 57, no. 3, pp. 140-151, June 2009.

[15] 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, 2011.

[16] H. C. Wu, N. I Wu, C. S. Tsai, M. S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," *IEE Proceedings Vision, Image and Signal Processing*, vol. 152, no. 5, pp. 611-615, Oct. 2005.

[17] N. I Wu, C. M. Wang, C. S. Tsai, M. S. Hwang, "A certificate-based watermarking scheme for coloured images," *The Image Science Journal*, vol. 56, pp. 326-332, 2008.

[18] G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map", *Pattern Recognition Letters*, vol. 31, pp. 347-354, 2010.

[19] S. Zhang, J. Chen, "Digital image scrambling technology based on matrix transformation", *Journal of*

Fujian Normal University (Natural Science Edition), vol. 20, no. 4, pp. 1-5, 2004.

Dong Wang received his B.S. in Applied Mathematics from Xidian University in 1999; the M.S. in Applied Mathematics from Guilin University of Electronic and Technology in 2004. He is currently pursuing the Ph.D.degree in Applied Mathematics from Xidian University. His research interests include image processing and information security.

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. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. He is a Fellow of IEEE and a Fellow of IEE, UK. His research interests include database design, computer cryptography, image compression and data structures.

Yining Liu is currently an associate professor in Guilin University of Electronic Technology, Guilin, China. He is also a researcher in Guangxi Key Lab of Trusted Software. He received his BS degree in Applied Mathematics from Information Engineering University, Zhengzhou, China, in 1995; MS in Computer Software and Theory from Huazhong University of Science and Technology, Wuhan, China, in 2003; and PhD degree in Mathematics from Hubei University, Wuhan, China, in 2007. His research interests focus on the analysis of security protocols and secure e-voting.

Guoxiang Song was born in 1938, she is currently a professor of applied mathematics at Xidian University. Her research interests include numerical analysis, wavelets, and partial differential equations for image processing.

Yunbo Liu received his B.S. in Information and Computing Science from Guilin University of Electronic Technology in 2012. His research interests include image processing and information security.

An Efficient Approach for Privacy Preserving Distributed Clustering in Semi-honest Model Using Elliptic Curve Cryptography

Sankita J. Patel[1], Dharmen Punjani[2], and Devesh C. Jinwala[1]
(Corresponding author: Sankita J. Patel)
Department of Computer Engineering, S. V. National Institute of Technology[1]
Ichhchhanath, Surat, Gujarat, India
(Email: sankitapatel@gmail.com)
Department of Computer Science and Engineering, BITS Edu Campus, Vadodara, Gujarat, India[2]
(Received Sept 27, 2013; revised and accepted Mar. 25 & Dec. 2014)

Abstract

In this paper, we propose an approach that illustrates the application of Elliptic Curve Cryptography (ECC) in Privacy-preserving distributed K-Means Clustering over horizontally partitioned dataset. We believe that the conventional cryptographic approaches and secret sharing schemes for privacy-preserving distributed K-Means clustering, are not scalable due to the higher computational and communication cost. Elliptic Curve based cryptosystems offer much better key size to security ratio in comparison. Hence, we use ECC based ElGamal scheme in distributed K-Means clustering to preserve privacy. Our approach avoids multiple cipher operations at each site and hence is efficient in terms of computational cost. We also achieve a reduction in the communication cost by allowing parties to communicate in a ring topology. Our experimental results show that our approach is scalable in terms of dataset size and number of parties in a distributed scenario. We carry out comparative analysis of our approach with existing approaches to highlight the effectiveness of our approach.

Keywords: Elliptic curve cryptography, K-means clustering, privacy preservation in data mining, secure multiparty computation

1 Introduction

With the advancement in communication technologies, distributed database applications have become commonplace. One of the inherent components of distributed database applications is distributed data mining, used to extract meaningful information from distributed databases. However, distributed data mining is an operation that demands sharing of data amongst organizations.

Such sharing though is necessary, poses threat to the privacy concerns of the organization's data. For example, consider a typical medical research application in which, a number of different hospitals wish to mine the patient data cooperatively and collaboratively. However, privacy policies and laws prevent these hospitals from over pooling their data or revealing it to each other. In such cases, classical data mining solutions cannot be used. Rather, it is necessary to find a solution that enables the hospitals to mine the desired data on the union of their databases, without ever pooling or revealing their individual data. In addition, in order to address the privacy concerns about the patient's data genuinely, appropriate privacy-preserving mechanisms must also be incorporated with such data mining algorithms. The focus of this paper is broadly on Privacy Preserving Distributed Data Mining (PPDDM).

PPDDM has indeed been addressed by the data mining and security researchers in the past decade. As we further discuss in Section 8, broadly, two approaches for the purpose, have been advocated in the literature viz., the *randomization based* and the *cryptography based*.

Our focus here is on cryptographic approaches owing to them offering high accuracy in privacy preservation and mining [30]. However, we also observe that the protocols proposed therein are expensive in terms of computational and communication overheads and hence are not scalable [29]. Therefore, the chief concern in designing such protocols must be on minimizing the overheads incurred.

In this paper, we focus on the issue of minimizing the overheads associated with cryptography based PPDDM approaches. We propose a privacy-preserving distributed K-Means clustering algorithm for the purpose. We justify our approach using theoretical analysis and empirical evaluation. As we prove formally in Section 5 that our approach preserves privacy in a semi-honest model; whereas our empirical observations discussed in Section 7, show

Table 1: The state-of-the-art in privacy preservation in clustering

Sr. No.	Privacy Preserving Clustering approaches	Privacy	Overheads	Remarks
1	Randomization based [25]	Low	Low	-Less accuracy
2	Oblivious transfer based [34, 13, 14, 5]	High	High	-Poor scalability in terms of dataset size and number of parties
3	Homomorphic encryption based [14, 15]	High	High	-Poor scalability in terms of dataset size and number of parties
4	Secret sharing based [33, 6, 27, 28]	High	Low	-Poor scalability in terms of number of parties -Use non colluding/trusted third parties

that our approach improves upon the existing approaches in terms of overheads entailed in communication and computational cost.

As we show in Table 1, there are four main approaches for privacy preservation in clustering viz., the *randomization-based*, the *oblivious transfer-based*, the *homomorphic encryption-based* and the *secret sharing-based*.

We observe that the randomization-based approaches inherently offer lower confidence and hence are not suitable in privacy-critical applications [25]. The oblivious transfer based approach is not scalable due to its high overheads [5, 13, 14, 34] and hence we do not consider this approach in our subsequent discussion. Referring to the existing literature for homomorphic encryption based and the secret sharing based approaches, we observe the following key issues:

- Existing homomorphic encryption based approaches use classical public key cryptosystems, that could lead to higher overheads [14, 15].
- The majority of the solutions assumes Trusted Third Party (TTP) [13, 33, 34]. There exist approaches that avoid TTP though, they are costly in terms of communication overheads [6, 27, 28].
- There is a scope for improvement in the existing secret sharing based approaches in terms of reducing the communication cost to enable them scale to a number of parties in distributed scenario [6, 27, 28, 33].

In this paper, we collectively address the above issues using Elliptic Curve Cryptography (ECC).

ECC is an attractive approach that substitutes classical public key cryptosystems. This is due to the higher per bit security offered by ECC [17, 24]. In this paper, we discuss the application of ECC based ElGamal [8] scheme in privacy preserving distributed clustering. In addition to this, by setting up ring topology among the parties, we show the reduction in the communication cost. In addition, in the proposed approach, the parties do not rely on the TTP. This is desirable, since in a practical scenario, TTPs are hard to find and if found, a compromise in TTP will jeopardize the privacy of the entire protocol.

We illustrate the effectiveness of our algorithm by comparing it with secret sharing based approach [27] and encryption based approach [15]. We prove the security of our approach in a semi-honest model and show that our approach privately computes final cluster means without revealing local cluster means of each party.

Our contributions in this paper are following:

- We propose privacy preserving distributed K-Means clustering using Elliptic Curve Cryptography.
- We avoid the Trusted Third Party by using a secure multiparty computation protocol.
- We achieve communication efficiency of $O(N)$ (where N being the number of parties) in our protocol by allowing parties to communicate in a ring topology. Our theoretical analysis presented in Section 6 justifies this.
- To show the practical feasibility of our approach, we carry out empirical analysis in a distributed scenario with real datasets.

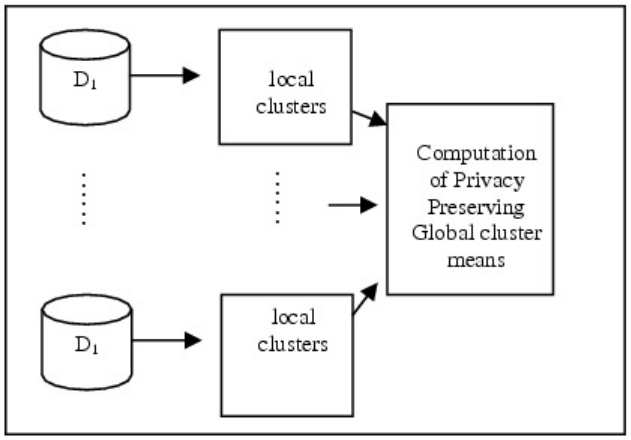


Figure 1: Distributed clustering

- We prove the privacy and correctness of our proposed approach.

2 The Preliminaries

2.1 Distributed K-Means Clustering

K-Means clustering aims to partition n records into k clusters in which each record belongs to the cluster with the nearest center [9, 22, 23]. Given an initial set of cluster means, the algorithm proceeds by alternating between two steps:

- **Assignment step:** For each record, calculate the distance of a record to all k means and assign record to the cluster with the closest mean.
- **Update step:** Calculate the new means to be the centroid of the records in the cluster.

However, in case of distributed dataset, the update step needs modification. This is since we need to calculate global clusters considering union of data located at each party. For distributed clustering, we follow approach of [15] to calculate the global clusters as shown in Figure 1.

Let us consider the dataset D=D_1,D_2,..,D_n, consisting of n records and each record D_i consists of m attributes. We assume horizontal partitioning of dataset D among N parties in which each party holds a subset of records of D with all m attributes. Each party first generates local clusters by applying K-Means clustering on data D_i. Then parties calculates sum of records and number of records in each cluster using the steps described in Algorithm 1.

Let us have k=2 clusters. Suppose the sum of the records in first and second clusters for party P_i are m_1^i and m_2^i and the numbers of records in the first and the second cluster are n_1^i and n_2^i respectively. Here, m_1^i and m_2^i are m dimensional vectors. Then global cluster means

Algorithm 1 Computation of sum of records and number of records

n: number of records
k: total number of clusters
$\mu_1, .., \mu_k$: initial clusters
1: Initialize n, k, $\mu_1, .., \mu_k$
2: do classify n records according to nearest μ
3: **for** $i = 1$ *to* k **do**
4: Let K_i be the i^{th} cluster
5: Compute $m_i = \sum_{x_j \in K_i} x_j$ and $n_i = |K_i|$
6: Output $(m_i,\ n_i)$
7: **end for**

considering union of data of N parties can be computed as,

$$\sum_{i=1}^{N} m_1^i / \sum_{i=1}^{N} n_1^i \ and \ \sum_{i=1}^{N} m_2^i / \sum_{i=1}^{N} n_2^i. \quad (1)$$

Particularly, we need to compute $(sum\ of\ records)/(number\ of\ records)$ for each cluster in a privacy preserving way. This ensures that the local cluster means are not revealed to other parties while computing global cluster means. This is the focus of this paper.

2.2 Homomorphic Encryption

A homomorphic encryption scheme allows certain algebraic operations to be carried out on the encrypted plaintext, by applying an efficient operation to the corresponding cipher-text. In additive homomorphic encryption schemes, the message space is a ring. Here, there exists an efficient algorithm whose input is the public key of the encryption scheme and two ciphertexts, and whose output is $E_{pk}(m_1) +_{pk} E_{pk}(m_2) = E_{pk}(m_1 + m_2)$. It is easy to compute, given the public key alone and the encryption of the sum of the plain texts of two cipher-texts. There can also be an efficient algorithm $\bullet_{pk}$, whose input consists of the public key of the encryption scheme, a cipher-text, and a constant c in the ring, and output is $c \bullet_{pk} E_{pk}(m) = E_{pk}(c \cdot m)$ [21].

2.3 Elliptic Curve Cryptography (ECC)

ECC was suggested independently by Koblitz [17] and Miller [24]. It is a public key approach that is defined on the algebraic structure of elliptic curves over finite fields. Elliptic Curves are defined over two types of finite fields: prime field F_p, where p is a large prime number, and binary field F_2^m. In this paper, we use elliptic curves over prime fields i.e. $E(F_p)$. $E(F_p)$ is defined as the set of solutions $(x, y) \in F_p \times F_p$ to Equation (2).

$$y^2 = x^3 + ax + b;\ a, b \in F_p. \quad (2)$$

ElGamal over Elliptic Curves

Applications of ElGamal encryption scheme has been discussed in [12, 19]. In this paper, we use ECC based ElGamal scheme to preserve privacy in distributed K-Means clustering. The original ElGamal encryption scheme [8] is multiplicatively homomorphic. In order to get the desired additive homomorphic property, it can be implemented in elliptic curves [20]. The methods for EC-ElGamal encryption and decryption are discussed in [32] and described in Algorithm 2.

Algorithm 2 Elliptic curve ElGamal encryption scheme

Private Key: $x \in F_p$
PublicKey: E,p,G,Y; whereby $Y = xG$ and elliptic curve E over F_p with G,$Y \in E$
Encryption: For a given plaintext $m \in [0, p-1]$ and random $k \in [1, n-1]$; where n is the order of E
M = Scalar_To_Point(m)
ciphertext C = Encryption(m) = $(R, S) = (kG, M + kY)$
Decryption: M = Decryption(C) = dec(R,S) = $-xR + S$
m =Point_To_Scalar(M)

The function Scalar_To_Point() converts the scalar values $m_i \in F_p$ into elliptic curve points $M_i \in E$; whereas the function Point_To_Scalar() does the reverse mapping from point to scalar value. The Scalar_To_Point function applied on scalar value m generates point M on the elliptic curve such that $M = mG$; where G is the generator point. The Scalar_To_Point() function satisfies the required homomorphic property, because,

$$\begin{aligned} & M_1 + M_2 + \ldots + M_n \\ = \; & Scalar\_To\_Point(m_1 + m_2 + \ldots + m_n) \\ = \; & (m_1 + m_2 + \ldots + m_n)G \\ = \; & m_1G + m_2G + \ldots + m_nG \end{aligned} \quad (3)$$

holds; where, m_1,m_2,$\ldots m_n \in F_p$ [32]. For the subsequent discussion, we call Scalar_To_Point() operation as mapping operation and Point_To_Scalar operation as remapping operation.

3 The Proposed Approach

In this paper, we focus on privacy preservation scheme for distributed K-Means clustering. As discussed in Section 2.1, distributed K-Means clustering requires computation of (sum of records)/(number of records) as an intermediate step to compute global cluster means in each iteration. However, for this computation, each party needs to share the pair (sum of records, number of records) with other parties. If these pairs are sent in clear, then there is a threat to privacy violations of these data. Hence, there is a need to share this data in such a way that the privacy is preserved and at the end, every party is able to get sum of the data.

We assume communication between parties in ring topology where each party is able to communicate with the next party in a ring. Communication in a ring topology avoids transmission of multiple messages among parties and thus reduces communication cost. The distribution of data among the parties is in horizontal partitioning model. We assume the semi-honest behavior of all parties i.e. all parties correctly follow the prescribed protocol.

We utilize ECC based ElGamal (EC-ElGamal) scheme to incorporate privacy in our protocol. The domain parameters for EC-ElGamal scheme are (a, b, p, G, n, E), where E is an elliptic curve defined over finite field F_p; where p is a large prime. a and b are coefficients of E that satisfy Equation (2). G is the base point of order n, and $nG = \infty$. The private key x is randomly selected from $[1, p-1]$ and the public key is $Y = xG$, another point on the curve.

Our protocol requires a pre-processing step that each party has to carry out. All parties have to generate data points on the elliptic curve and store them for future mapping or remapping purpose. However, this is the one time cost in the protocol.

The protocol starts with the party designated as initiator party in the ring. In fact, any party can act as an initiator. We assume the first party as an initiator party for the subsequent discussion. Communication in a ring topology restricts parties to communicate only with its immediate next neighbor. To give an example of N=3 party scenario, Party 1 can send data to Party 2 and receive data from Party 3, Party 2 can send data to Party 3 and receive data from Party 1 and Party 3 can send data to Party 1 and receive data from Party 2.

Our protocol works in two phases; in the first phase, each party performs local clustering and computes sum of records and number of records values for each cluster using steps described in Algorithm 1. Each party maps the values to the point on the elliptic curve. Initiator party, then encrypts the mapped values by adding noise to them. For that, initiator party selects random value k from $[1, n-1]$. Let us assume that one of the values is m and it is mapped to the point M on the elliptic curve using Equation (4).

$$M = mG, \quad (4)$$

where G is the base Point of the Elliptic Curve E. Now initiator party encrypts M and generates another point C on the elliptic curve using Equation (5).

$$C = M + kY. \quad (5)$$

This process is repeated for all values of sum of records and number of records for each cluster. Initiator party, then sends these values to the next party in the ring. The rest of the parties just adds their data points to the received value and sends this addition to next party in ring. This process is shown in Phase I in Figure 2.

Phase II of the protocol starts upon receiving message by the initiator party. If we consider three party scenario,

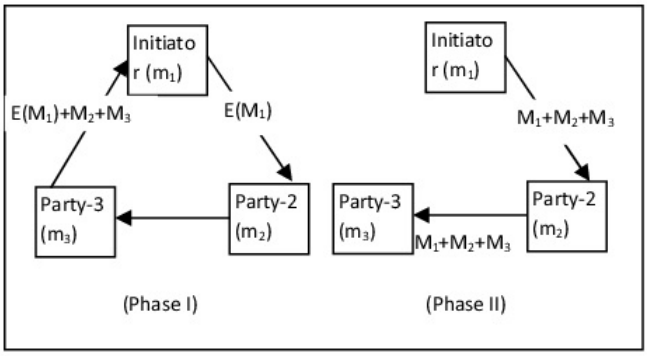


Figure 2: Secure addition using ECC

this message is $(E(M_1)+M_2+M_3)$, where m_1, m_2 and m_3 are the original data values of Party 1 (Initiator), Party 2 and Party 3 respectively and M_1, M_2 and M_3 are the corresponding points of Elliptic Curve. Upon receiving this message, initiator removes the noise added to the message during Phase I and gets $M_1 + M_2 + M_3$ as shown in Equation (6).

$$\begin{aligned} SUM &= (E(M_1) + M_2 + M_3) - kY \\ &= M_1 + kY + M_2 + M_3 - kY \\ &= M_1 + M_2 + M_3. \end{aligned} \quad (6)$$

Initiator forwards this sum to the next party in the ring and subsequently all parties receive the sum. As the sum is a point on the curve, all parties remap this sum and get the original sum value $m_1 + m_2 + m_3$. This process is repeated for all sum of records and number of records values and finally at the end of the iteration, all parties will be able to compute (sum of records)/(number of records) and hence global cluster means. The process of finding the global cluster means is repeated until the convergence criteria is met for K-Means clustering. The protocol is described in Algorithm 3.

Algorithm 3 Privacy preserving distributed K-means clustering using elliptic curve cryptography

P : Set of parties $\{P_1, P_2, \ldots, P_N\}$; where P_1 is the initiator
$v_{is} = (m_i, n_i)$: Secret value of party P_i, where m_i is sum of records and n_i is number of records in cluster i
Private Key of Initiator: $x \in [0, n-1]$; where n is the order of elliptic curve E
Public Key: E, p, G, Y whereby $Y = xG$ and elliptic curve E over F_p with G,$Y \in E$
M= Scalar_To_Point(m) as described in Algorithm 2
k : random value where $k \in [1, n-1]$
c : number of clusters
$\{\mu_1, \ldots, \mu_c\}$: set of initial cluster centers

1: **Do** in parallel for each party $P_i \in \{1 \ldots N\}$
2: find $((m_1, n_1), \ldots, (m_c, n_c))$ using Algorithm 1
3: map $(m_1, n_1), \ldots, (m_c, n_c))$ values on the curve E
4: **for** each μ_i, where $i \in 1$ *to* c **do**
5: **if** P_i is initiator **then**
6: encrypt the mapped values using equation C = Encryption(m) = (R,S) = (kG,$M + kY$)
7: Send it to the next party in ring
8: **else**
9: Add the mapped value to received point using the equation $SUM = M + C$
10: Send SUM to the next party in ring
11: **end if**
12: **if** Initiator received data **then**
13: remove the noise in data using the equation M = SUM-kY
14: remap M using the equation m = Point_To_Scalar(M); as described in Algorithm 2
15: send M to the next party in ring
16: **else**
17: receive M and send it to next party in ring if not last in the ring; remap M and get the sum
18: **end if**
19: recompute μ_i using received values
20: until termination criteria met
21: **end for**

4 Correctness Claim

In this section, we prove that our protocol correctly computes the global cluster means.

Lemma 1. *Privacy Preserving Distributed K-Means Clustering using ECC correctly computes global cluster means from the local cluster means of each party.*

Our protocol requires computation of intermediate cluster means from local cluster means generated at each intermediate step. This requires computation of (sum of records)/(number of records) values. These values are added using the secure addition protocol discussed in Section 3. With the help of a small example, we prove that our protocol computes the addition of values correctly and hence the final (sum of records)/(number of records) are computed correctly and subsequently the final cluster means.

For the sake of understanding, we take small parameters in the example. In practice, large parameters are used.

Let us consider the domain parameters (a, b, p, G, n, E), where p=11, a=1, b=6. The elliptic curve E over F_{11} is represented by the equation $y^2 = x^3 + x + 6$ mod 11. Points on this curve are (2,4), (2,7), (3,5), (3,6), (5,2), (5,9), (7,2), (7,9), (8,3), (8,8), (10,2), (10,9), along with a point at infinity ∞. The point G= (2, 7) is the generator point. The group of points generated by G over $E(F_{11})$ can be calculated using the equations of addition and doubling. The points are: G=(2,7) $2G$=(5,2), $3G$=(8,3), $4G$=(10,2), $5G$=(3,6), $6G$=(7,9), $7G$=(7,2), $8G$=(3,5), $9G$=(10,9) $10G$=(8,8) $11G$=(5,9) $12G$=(2,4). So the order n is equal to 13 because $nG = \infty$. Assume the private key x=6, we obtain the public key point

$Y = xG = 6G$ = (7,9). Let us assume three party protocol where the private values of Party 1,2 and 3 are 5, 3 and 2 respectively. The point representation of these values are M_1=5G=(3,6), M_2=3G=(8,3) and M_3=2G=(5,2) respectively. Let us have x=6 and hence Y=xG=(7,9). Assume the value of k=4.

Message generated by Initiator = $M_1 + kY = (3,6) + 4(7,9) = (8,3)$.

Party 2 receives message from initiator party and adds its value M_2 to received message. Hence, the message generated by Party 2 = (8,3) + (8,3)=(7,9). Party 2 sends this message to Party 3. Party 3 does the same processing and computes the message (7,9) + (5,2) = (3,5) and send it to initiator party.

Upon receiving message from Party 3, initiator party removes the noise from received message by computing : $(M_1 + k_Y + M_2 + M_3)$ kY = (3,5) + (5,-9) = (3,9) + (5,2) = (8,8) = 10G. Hence, the computed sum is 10. Similarly value (8,8) is forwarded to rest of the parties in ring and remaping the same yields the sum value.

After computing sum of records and number of records securely, parties compute global cluster means by computing ((sum of records)/(number of records)) and subsequently the final cluster means.

5 The Security Analysis

5.1 Security Model

To prove that our proposed approach computes the clusters for every record in a privacy preserving way, we first need to define privacy. For that, we use the concept of Secure Multiparty Computation that is defined in [10].

Set Up Assumptions.
We assume the existence of a public key infrastructure, where each party knows ECC parameters corresponding to each of the other parties. Communication between the parties is in a ring topology and one party is randomly designated as initiator.

Communication Channel.
We assume that the adversary can tap all the communication channels i.e. the channels do not provide privacy. In addition to this, the adversary can't omit, replay, duplicate or generate messages over the communication channel; i.e. the channels are postulated to be reliable. Moreover, communication channel is synchronous and point-to-point link exists between communicating parties. However, in this paper, we assume that the parties communicate with each other in ring topology i.e. a point-to-point link exists between each party and its two neighbors.

Adversary Model.
We assume a semi-honest adversary model. A semi-honest party correctly follows the protocol using its correct input. However, such party can compromise the security by later using the messages exchanged during the execution of the protocol. On contrary, the malicious party disrupts the protocol by deviating from the specified program. Whereas the malicious behavior may not be feasible to many users, the semi-honest model may be feasible to them. This is because deviating from a specified protocol is more difficult than merely recording some communication registers [10].

Computational Power.
We consider computationally bounded semi-honest adversaries.

5.2 Proof of Privacy

We first present the high level argument for how our protocol protects each party's private data. As discussed in Section 3, key generation is performed by the initiator party and it generates private key x and corresponding public key Y=xG. The initiator party encrypts messages by adding the randomness kY in the message (i.e. $M + kY$; where M is the point representation of message m). All other parties add their messages in the received message.

To make this notion more formal, we start with the formal definitions of security in semi-honest model. In fact, our proposed approach requires secure multiparty addition at intermediate step and this is the only step that concerns with privacy. Parties locally carry out the rest of the steps. Hence, we use the framework defined for secure multiparty computation by Goldreich [10]. We use the formulation of a semi-honest model that is a straightforward extension of definition of zero knowledge and prove the security in this model. For simplicity, we use the two-party case here. Same with minor modifications can be applied to multiparty case. More details may be found in [10].

According to Goldreich [10], *"a protocol privately computes function f, if whatever can be obtained from a party's view of semi-honest execution, could be essentially obtained from the input and output available to that party".*

Definition 1. *Let $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^* \times \{0,1\}^*$ be a functionality, and $f_1(x,y)$(resp. $f_2(x,y)$) denotes the first(resp. second) element of $f(x,y)$. Let Π be a two party protocol for computing f. The view of the first(resp. second) party during the execution of Π on (x,y) denoted as $VIEW_1^{\Pi}(x,y)$ (resp. $VIEW_2^{\Pi}(x,y)$), is (x,r,(m_1,m_2, ...,m_t)) (resp. (y,r,(m_1,m_2,...,m_t))), where r represents the randomness of first(resp. second) party and m_i represents the i_{th} message it has received. The output of the first party after the execution of Π on (x,y), denoted $OUTPUT_1^{\Pi}(x,y)$ (resp. $OUTPUT_2^{\Pi}(x,y)$) is implicit in the party's own view of execution, and $OUTPUT^{\Pi}(x,y)$=($OUTPUT_1^{\Pi}(x,y)$, $OUTPUT_2^{\Pi}(x,y)$).*

We say that Π privately computes f, if there exists probabilistic polynomial time simulators S_1 and S_2 such

that,

$$\begin{aligned} & \{S_1((x, f_1(x,y)), f(x,y))\}_{(x,y)} \\ \equiv \quad & \{VIEW_1^{\Pi}(x,y), OUTPUT^{\Pi}(x,y)\}_{(x,y)} \\ & \{S_2((y, f_2(x,y)), f(x,y))\}_{(x,y)} \\ \equiv \quad & \{VIEW_2^{\Pi}(x,y), OUTPUT^{\Pi}(x,y)\}_{(x,y)}, \end{aligned}$$

where $\equiv$ *denotes computational indistinguishability.*

In addition to this, we formally define discrete logarithm problem for elliptic curves on which the security of our scheme relies.

Definition 2. *Elliptic Curve Discrete Logarithm Problem (ECDLP). Given an elliptic curve E defined over* F_p *and two points P,Q* $\in E$*, find an integer x such that* $Q = xP$ *if such x exists. This problem is more intractable than the classical Discrete logarithm problem [17].*

Definition 3. *Elliptic Curve (computational) Diffie Hellman Problem (ECDHP). Given an elliptic curve E defined over* F_p *and the points P, xP and* $yP \in E$*, compute the point xyP.*

Definition 4. *Elliptic Curve (computational)Decisional Diffie Hellman Problem (ECDDHP). Given an elliptic curve E defined over* F_p *and the points P, xP, yP and* $zP \in E$*, determine whether* $zP = xyP$*.*

ECDHP and ECDDHP polynomial time reduce to ECDLP [18]. In [4], authors prove that if ECDLP can not be solved in polynomial time, then neither can ECDHP and ECDDHP.

With the above definitions, we now prove the privacy of our protocol.

Lemma 2. *Privacy Preserving Distributed K-Means Clustering using ECC leaks no information beyond the sum of the secret values and the final cluster centers.*

Suppose S be a set of parties in a protocol where $S=\{1, 2, \ldots, n\}$, M_i, where $i \in S$ are the messages sent by $Party_1, Party_2, \ldots, Party_n$ respectively during first phase of the protocol. Let $i = 1$ be the initiator party in the protocol. The view of the party following the initiator party (i.e. party $i = 2$) and the view of rest of the parties is different.
We define the view of the party $i = 2$ as,

$$VIEW_i = M_{i \in initiator}.$$

We define the view of rest of the parties as,

$$VIEW_{i \in \ rest\ of\ the\ parties} = M_{i-1}.$$

For any S of size $(n-1)$ and any $VIEW_s$,

$$Prob[D\ has\ secret\ s\ |VIEW_s] =$$

$$Prob[D\ has\ secret\ s]\ for\ all\ s \in F_p.$$

Interested readers may refer Goldreich [10] for proof and other details.

First, the initiator party encrypts its message by adding the randomness in the message in form of kY, where k is the random value. From the view of the party receiving message from initiator party, it is clear that the party will not be able to decrypt the message without knowing k value. Accordingly, the privacy of initiator party's data is preserved.

Second, rest of the parties (i.e. all the parties except the one that receives data from initiator) adds its message to the received message. The message generated thus becomes random value and according to the view of the party receiving this message, party will not be able to decrypt the message due to ECDLP described in Definition 2.

During the second phase of the protocol, the initiator party extracts the sum of secret values and forwards it to all parties in a ring. Parties remap the sum, get the actual value of sum and consequently compute the cluster means.

In addition, let us consider the view of an adversary,

$$VIEW_{adversary} = M_i;\ where\ i \in \{1, 2, \ldots, n\}.$$

Let us consider the first phase of the protocol. Since, each M_i contains the randomness kY (which is added by the initiator party), even though the adversary gets messages M_i (by tapping the communication channels between parties); no private value is disclosed without knowing k value.

However, if adversary taps the communication channel between party i and $(i+1)$ and party $(i+1)$ and $(i+2)$; the adversary will be able to get the message M_{i+1}. Since, the message is a point on elliptic curve, adversary will not be able to get the plaintext message due to ECDLP described in Definition 2.

Similarly, by tapping all communication channels, the adversary will be able to get messages $M_i s$. However, getting such multiple messages do not reveal any information about the secret values due to the ECDHP and the ECDDHP described in Definitions 3 and 4, respectively.

6 Cost Analysis

6.1 Computational Cost

The computational cost of our protocol depends on the selection of initial clusters and the number of iterations required for converging the algorithm. For the fair analysis, we give here the computational cost for a single iteration. The computational cost is due to the operations in first round where parties send data to the next party in ring. However, we need to consider the cost for initiator and the rest of the parties separately. This is because the initiator encrypts and decrypts the data and the rest of the parties just need to add their data in the received data. Suppose to add the noise k, time taken is t_1. It is pointed out in [17] that by means of $O(logk)$ addition and doubling, one can compute kM value. Hence t_1 becomes $O(logk)$. Further,

to remove the noise, time taken is t_2. Let us have the time taken by other parties to add their data point is t_3. If we have r number of attributes and c number of clusters, the computational cost for single iteration for initiator can be $(O(logk)+t_2)*r*c = O(rclogk)$. For the rest of the parties the cost becomes $t_3 * r * c$ and thus $O(rc)$. For total of N parties the cost becomes $O(rclogk) + O(Nrc)$.

6.2 Communication Cost

Our protocol involves two rounds of communication per iteration. In first round, the sum of records and number of records in the cluster as elliptic curve points are communicated to the next party in the ring. If we have r number of attributes and total c clusters, we need to send $(r * c + c)$ elliptic curve points to the next party in first round of communication. In second round, only the sum (that is calculated) by initiator party is communicated to all parties in a ring. For r attributes and c clusters, each party needs to send $r * c$ values. Thus for N parties in the protocol, the communication cost becomes $O(Nrc)$ i.e. linear in terms of number of parties.

Comparison with Existing Approach
We compare the communication cost of our approach with the approach proposed [27] as our approach is close to [27]. For the subsequent discussion, we call approach of [27] as PPKMeansShamir and our approach as PPKMeansECC. The similarity between PPKMeansECC and PPKMeansShamir is that both propose secure multiparty addition as basic construct to perform Privacy Preserving Distributed K-Means Clustering. Both assume horizontal partitioning of data and the semi-honest adversary model. However, PPKMeansShamir utilizes Shamir's secret sharing scheme to preserve privacy while PPKMeansECC utilizes EC-Elgamal scheme to preserve privacy. However, we achieve improvement in the communication efficiency over PPKMeansShamir. To justify this, we theoretically analyze the number of messages exchanged during the single instance (i.e. for sending single value) of secure addition protocol. We assume distributed setup of N parties.

- PPKMeanShamir requires two rounds of operation. In first round, every party generates and distributes secret values i.e. sum of records and number of records to all parties. In second round, each party distributes the sum of shares to all parties. Hence, for each party the communication cost becomes $2(N-1)$ messages and for N parties the cost becomes $2N(N-1)$ and hence $O(N^2)$.
- PPKMeansECC requires two rounds of operation similar to PPKMeansShamir. As discussed in Section 3, we set up the ring topology for the communication among parties. Hence, in first round each party sends a message only to its neighbor. However, due to the ECC based approach, each party needs to send a point on curve and hence two messages. So total N points and $2N$ messages are communicated in first round. In second round, the initiator party decrypts the sum value and sends it to its neighbor that will then be forwarded by all other party except the last in ring. Therefore, in second round, the total messages are $(N-1)$. Hence, the total communication cost for N parties becomes $2N+(N-1)$ messages and hence $O(N)$.

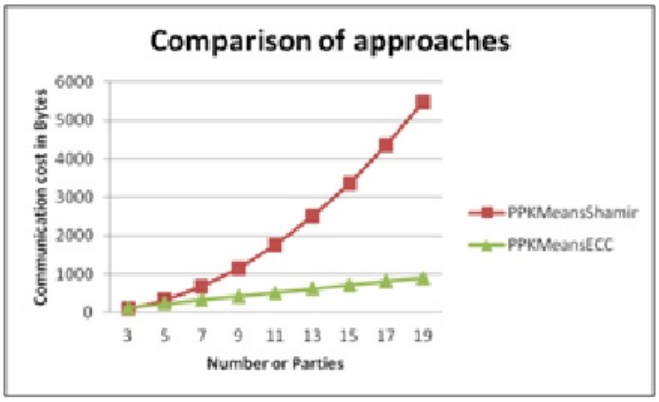


Figure 3: Comparison with respect to communication cost

Now, considering the above cost, we show the comparison of PPKMeansECC with PPKMeansShamir. We consider the communication cost per attribute per cluster per iteration in order to make it generalized. For PPKMeansShamir, we consider long integer values for shares while for our protocol, we consider 160-bit EC-ElGamal scheme and hence each point represents two coordinates of 160-bits each. The graph in Figure 3 shows the comparison.

7 Implementation and Experimental Evaluation

7.1 Experimental Setup

We carry out empirical analysis of our proposed algorithm in JAVA. The experiments are conducted on three different machines to emulate the distributed scenario consisting of three parties. All machines have similar configuration of Intel Core i5 processor, 4GB of RAM and 3.20GHz of processing power.

All the participating parties initially agree upon the cluster centers that are selected randomly and ECC parameters. The ring topology among the parties is set up and each party knows its neighbor on a ring. One party in a ring is randomly designated as Initiator.

We assume horizontal partitioning of datasets. In order to show realistic results, we test our distributed application on real datasets. Three datasets are taken from UCI machine learning repository. We give brief outline of the datasets here. Dataset1 is the Haberman's Survival

Table 2: Increase in computational cost and communication cost with respect to distributed K-means clustering

Party	Datasets	Accuracy		Number of iterations		Time (milliseconds)		Number of sending operations	
		DKM	Our approach	DKM	Our approach	DKM	Our approach	DKM	Our approach
P_1	Haberman's survival	50	50	5	5	5064	7443	10	10
P_2	Haberman's survival	63.68	63.68	5	5	3958	5361	10	10
P_3	Haberman's survival	50	50	5	5	2722	4972	5	5
P_1	STATLOG	61.51	61.51	16	16	4492	12912	32	32
P_2	STATLOG	61.11	61.11	16	16	3448	5218	32	32
P_3	STATLOG	52.68	52.68	16	16	2149	11250	16	16
P_1	SPECTF Heart	67.95	67.95	3	3	4382	5172	6	6
P_2	SPECTF Heart	65.33	65.33	3	3	2462	3881	6	6
P_3	SPECTF Heart	62.81	62.81	3	3	2198	3099	3	3

dataset consisting of 306 data records with 3 attributes and 2 class labels [1]. Dataset2 is the StatLog dataset consisting of 4435 data records with 36 attributes and 6 class labels [3]. Dataset3 is the SPECTF Heart consisting of 267 data records with 44 attributes and 2 different class labels [2]. To test the distributed application over horizontally partitioned data on three parties, we divide the available datasets into three subsets where each subset contains a partial set of records with all the attributes. We place these three data sets on three different machines to perform real time distributed clustering. Our test application successfully shows fully functional distributed clustering over real network.

7.2 Experimental Results

We evaluate our approach based on three metrics viz., *accuracy*, *computational cost* and *communication cost*. Accuracy is measured in percentage of records that fall into the cluster similar to their class labels. Computational cost is measured in terms of time taken for execution of the protocol and communication cost is measured in terms of the number of messages exchanged over communication channel.

Table 2 shows the increase in computational cost and communication cost with respect to distributed K-Means clustering without privacy preservation. Due to the random selection of initial cluster centers, we do not get same results for same dataset every time. Hence, we run our experiments five times and measure the average time and cost of the protocol from these experiments. Table 2 shows these results. In Table 2, P_1, P_2 and P_3 denote Party 1 (Initiator), Party 2 and Party 3 respectively. We show communication cost in terms of number of sending operations.

As shown in Table 2, we achieve similar level of accuracy even after incorporating privacy. Our proposed approach runs within 3-9 seconds of difference compared to the distributed k-means clustering without privacy preservation. This is due to the elliptic curve based addition and scalar multiplication operations carried out by parties.

To compare the computational cost of our proposed approach, we consider results of homomorphic encryption based approach proposed in [15] and PPKMeansShamir proposed in [27]. However, in these approaches, results are given for river dataset consisting of 25 KB size and 15 attributes which is different from datasets we have taken for experiments. Further, in [15], results are given for two party case. We consider three party case. As discussed in Section 6, the computation cost depends on number of attributes and not on number of records. For this reason, we consider the results of StatLog dataset consisting of 36 attributes to get the reasonable comparison. If we ignore the datasets and simply take per iteration per attribute statistics, we get the results shown in Table 3.

As shown, compared to homomorphic encryption based

Table 3: Comparison in computational cost of our approach with PPKMeansShamir and approach of [15]

Approach	**percentage increase in time over distributed K-Means clustering without privacy preservation (milliseconds)**
PPKmeans using homomorphic Encryption [15]	4915%
PPKMeansShamir [27]	25.26%
Our Protocol	49.30%

approach proposed in [15], our approach performs better in terms of computational cost. The reason is straightforward due to ECC based encryption technique. However, compared to PPKMeansShamir our approach incurs more computational cost. This is because; PPKMeansShamir requires only primitive operations on polynomials such as multiplication and interpolation. Whereas our approach requires cipher operations such as point addition and multiplication and map and remap operations.

Number of sending operations remains same for Distributed K-Means clustering and our protocol. However, in distributed K-Means clustering only small values are sent across the network. Whereas in our protocol, all values are sent as elliptic curve points and for 160-bit EC-Elgamal, it reaches 40 bytes per message.

8 Related Work

The review of state of the art methods for PPDM suggests two categories of PPDM approaches: 1. Randomization Based and 2. Cryptography Based [29]. The Randomization Based approach, though efficient, provides a lower level of privacy [29]. The Cryptography based approach achieves higher level of privacy but at the cost of higher computation and communication overheads [29]. Hence, the research in cryptography based approach mainly concentrates on reducing these overheads.

Various approaches to preserve privacy in association rule mining [7] and classification [35] are proposed. However, in this paper, we investigate a novel cryptography based approach for clustering that is efficient in terms of computational and communication overheads.

The state-of-the-art in privacy preserving clustering suggests three categories of cryptography based approaches such as *oblivious transfer* based, *homomorphic encryption* based and *secret sharing* based. Oblivious transfer based approaches for clustering are proposed in [5, 13, 14, 34]. The limitation of these approaches is that they are computationally expensive and hence their scope is limited to small datasets only. As compared, homomorphic encryption based approach provides higher level of privacy but incurs higher computational cost. This is because existing approaches [14, 15] in this category uses classical public key cryptosystems. Authors in [5] and [15] address privacy-preserving clustering for arbitrarily partitioned and horizontally partitioned data respectively. However, they only present two party case for semi-honest model. In practical scenario, it is common to have more than two parties in distributed applications. Secret sharing based approach is an attractive solution to PPDM, which greatly reduces the computational cost of oblivious transfer and homomorphic encryption and provides higher level of privacy [29]. Privacy preserving clustering based on secret sharing has been addressed in [6, 27, 28, 33]. However, security in [33] and [6] is based on some form of trusted party. For example, [33] relies on cloud computing servers to compute clusters, whereas [6] assumes two non-colluding third parties to compute cluster means. However, trusted parties are hard to achieve in practical scenario and if achieved it becomes single point of failure if compromised. In addition to this, secret sharing based approaches incur higher communication cost. This is because the approach requires parties to send messages to every other party in the protocol. For example, [27] and [28] propose privacy preserving protocol for K-Means clustering using Shamir's secret sharing scheme. Though efficient in terms of computational cost, the communication complexity is $O(N^2)$ for N parties in distributed setup. This is not feasible for scenarios having numerous parties.

Elliptic Curve Cryptography is a major break-through in public key cryptography. However, a few approaches exist that actually implements ECC in privacy preserving data mining. Authors in [26] propose Privacy Preserving Association Rule Mining in Unsecured Distributed Environment Using ECC. However, the higher computational cost makes it impractical for large datasets. One more protocol for the association rule mining is proposed in [31]. This protocol is closest to our protocol. In that, authors use ECC to collect the local frequent itemsets of all the sites at the initiator. Initially, the site who wants to initiate the mining process encrypts all its local frequent itemsets and sends them to the next site. The next site encrypts all its local frequent itemsets along with its itemsets received from the initiator and pass it to its next site and so on. The procedure continues for all the sites. Once the turn reached to the initiator, initiator starts decrypting the encrypted message and passes it to

the next site. The process continues until the initiator site is reached. At this point, all global candidate itemsets are collected in the initiator site. However, multiple encryption-decryption at each site results in high computational cost and hence poor scalability.

Authors in [11] propose a protocol for performing Secure Union using common decryption key. The main emphasis in their approach is to optimize the performance of privacy preserving scheme for association rule mining. In [11] the application of ECC versus Exponential Cryptography is presented as well.

Recently, authors in [16] have proposed a framework using ECC for extremely secure transmission in distributed privacy preserving data mining. The proposed framework has two major tasks, secure transmission, and privacy of confidential information during mining. Secure transmission is handled by using ECC and data distortion for privacy preservation ensuring highly secure environment. However, data distortion results in lesser accuracy in data mining results. For the critical applications, accuracy of data mining results is utmost important.

9 Conclusions

Scalable techniques for privacy preservation are desirable for emerging distributed database applications consisting of large datasets and number of parties. Existing PPDM techniques are either computationally expensive due to classical public key cryptosystems or communicationally inefficient due to secret sharing based approach. In addition, the majority of the approaches uses some form of trusted third party. In this paper, we collectively address this issue and propose a novel approach to privacy preserving distributed K-Means clustering using Elliptic Curve Cryptography. Our approach improves on existing approaches in terms of computational cost by utilizing Elliptic Curve Cryptography. Setup of parties in ring topology helps us to achieve efficiency in terms of communication cost. Finally, we avoid use of trusted third party by implementing secure multiparty computation.

However, due to ring topology, one malfunctioning workstation can create problems for the entire network and hence our approach demands reliable channels. In future, we intend to explore and analyze better alternatives to EC-Elgamal scheme in our protocol.

References

[1] "Available: http://archive.ics.uci.edu/ml/machine-learning-databases/haberman,".

[2] "Available: http://archive.ics.uci.edu/ml/machine-learning-databases/spect/spectf.train,".

[3] "Available: http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/satimage,".

[4] D. Boneh and R. J. Lipton, "Algorithms for black-box fields and their application to cryptography," in *Advances in Cryptology (Crypto96)*, pp. 283–297, Springer, 1996.

[5] P. Bunn and R. Ostrovsky, "Secure two-party k-means clustering," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 486–497, 2007.

[6] M. C. Doganay, T. B. Pedersen, Y. Saygin, E. Savaş, and A. Levi, "Distributed privacy preserving K-means clustering with additive secret sharing," in *Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society*, pp. 3–11, 2008.

[7] A. El-Sisi and H. M. Mousa, "Evaluation of encryption algorithms for privacy preserving association rules mining," *International Journal of Network Security*, vol. 14, no. 5, pp. 289–296, 2012.

[8] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology*, pp. 10–18, Springer, 1985.

[9] E. W. Forgy, "Cluster analysis of multivariate data: Efficiency versus interpretability of classifications," *Biometrics*, vol. 21, pp. 768–769, 1965.

[10] O. Goldreich, *Foundations of Cryptography: Basic Applications*, vol. 2, Cambridge university press, 2009.

[11] M. Gorawski and Z. Siedlecki, "Optimization of privacy preserving mechanisms in homogeneous collaborative association rules mining," in *2011 Sixth International Conference on Availability, Reliability and Security (ARES'11),* pp. 347–352, 2011.

[12] 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.

[13] A. Inan, S. V. Kaya, Y. Saygın, E. Savaş, A. A. Hintoğlu, and A. Levi, "Privacy preserving clustering on horizontally partitioned data," *Data & Knowledge Engineering*, vol. 63, no. 3, pp. 646–666, 2007.

[14] G. Jagannathan and R. N. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 593–599, 2005.

[15] S. Jha, L. Kruger, and P. McDaniel, "Privacy preserving clustering," in *Computer Security – ESORICS'05*, pp. 397–417, Springer, 2005.

[16] P. Kiran, S. S. Kumar, and N. P. Kavya, "A novel framework using elliptic curve cryptography for extremely secure transmission in distributed privacy preserving data mining," *arXiv Preprint arXiv:1204.2610*, 2012.

[17] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[18] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," in *Towards a Quarter-Century of Public Key Cryptography*, pp. 103–123, Springer, 2000.

[19] C. C. Lee, M. S. Hwang, and S. F. Tzeng, "A new convertible authenticated encryption scheme based on the elgamal cryptosystem," *International Journal of Foundations of Computer Science*, vol. 20, no. 02, pp. 351–359, 2009.

[20] M. Leslie, "Elliptic curve cryptography," *An ECC Research Project*, 2006.

[21] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, p. 5, 2009.

[22] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory,* vol. 28, no. 2, pp. 129–137, 1982.

[23] J. MacQueen, et al., "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, California, USA, 1967.

[24] V. S. Miller, "Use of elliptic curves in cryptography," in *Proceedings of Advances in Cryptology (Crypto85)*, pp. 417–426, Springer, 1986.

[25] S. R. M. Oliveira and O. R Zaiane, "Privacy preserving clustering by data transformation.," in *SBBD*, pp. 304–318, 2003.

[26] A. C. Patel, U. P. Rao, and D. R. Patee, "Privacy preserving association rules in unsecured distributed environment using cryptography," in *the Proceedings of the Third International Conference on Computing Communication & Networking Technologies (ICCCNT'12)*, pp. 1–5, 2012.

[27] S. Patel, S. Garasia, and D. Jinwala, "An efficient approach for privacy preserving distributed K-means clustering based on shamirs secret sharing scheme," in *Trust Management VI*, pp. 129–141, Springer, 2012.

[28] S. Patel, V. Patel, and D. Jinwala, "Privacy preserving distributed k-means clustering in malicious model using zero knowledge proof," in *Distributed Computing and Internet Technology*, pp. 420–431, Springer, 2013.

[29] T. B. Pedersen, Y. Saygın, and E. Savaş, "Secret charing vs. encryption-based techniques for privacy preserving data mining," in *Joint UNECE/Eurostat Work Session on Statistical Disclosure Control*, 2007.

[30] B. Pinkas, "Cryptographic techniques for privacy-preserving data mining," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 2, pp. 12–19, 2002.

[31] M. Rajalakshmi and T. Purusothaman, "Privacy preserving distributed data mining using randomized site selection," *European Journal Of Scientific Research*, vol. 64, no. 2, pp. 610–624, 2011.

[32] O. Ugus, A. Hessler, and D. Westhoff, "Performance of additive homomorphic ec-elgamal encryption for tinypeds," *6. Fachgespräch Sensornetzwerke*, pp. 55, 2007.

[33] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar, "Efficient privacy preserving k-means clustering," in *Intelligence and Security Informatics*, pp. 154–166, Springer, 2010.

[34] J. Vaidya and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 206–215, 2003.

[35] J. Z. Zhan, L. W. Chang, and S. Matwin, "Privacy preserving K-nearest neighbor classification.," *International Journal of Network Security*, vol. 1, no. 1, pp. 46–51, 2005.

Sankita J. Patel is currently pursuing PhD from the Department of Computer Engineering, Sardar Vallabhbhai National Institute of Technology (SVNIT), Surat, India. She is an Assistant Professor at SVNIT, Surat. Her research interests include Information and Communications Security and Privacy and Privacy preservation techniques in data mining.

Dharmen Punjani is an Assistant Professor at BITS edu campus, Babaria Institute of Technology, Vadodara. His research interests include Privacy preservation techniques in data mining, Information security, BIG DATA analysis and data mining techniques.

Devesh C. Jinwala is working as a Professor at the Department of Computer Engineering, Sardar Vallabhbhai National Institute of Technology (SVNIT), Surat, India. He has been employed with SVNIT since Jan 1991. His research focus is primarily in Information Security and Privacy issues in Wireless Sensor Networks, Cryptography and Software Requirements Specifications.

An Improved Multi-receiver Generalized Signcryption Scheme

Cai-Xue Zhou
School of Information Science & Technology, University of Jiujiang, JiuJiang 332005, P. R. China
(Email: charlesjjjx@126.com)
(Received May 12, 2013; revised and accepted Mar. 6 & Nov. 3, 2014)

Abstract

Generalized signcryption (GSC) scheme can adaptively work as an encryption scheme, a signature scheme or a signcryption scheme with only one algorithm and one keypair. It can realize using the same keypair in more different cryptographic primitives. It is very suitable for storage-constrained environments, like the embedded systems, smart cards and wireless sensor networks. In this paper, we analyze a multi-receiver GSC scheme, and show that it cannot achieve indistinguishability-adaptive chosen ciphertext attack (IND-CCA2) secure in the pure encryption mode and hybrid encryption mode. We further propose a revised version of the scheme, which resolves the security issues of the original scheme without sacrificing its high efficiency and simple design. Our improved scheme can be proved to be IND-CCA2 secure and existentially unforgeable-adaptive chosen message attack (EUF-CMA) under computational Diffie-Hellman (CDH) assumption.

Keywords: Adaptive chosen ciphertext attack, adaptive chosen message attack, generalized signcryption, multi-receiver generalized signcryption, randomness reuse

1 Introduction

In Asiacrypt 2011, Paterson et al. [31] revisited the problem where a single keypair is used for both encryption and signature primitives. This usage can reduce storage requirements, the cost of key certification and the time taken to verify certificates. These savings may be critical in embedded systems and low-end smart card applications. As a prime example, the globally-deployed EMV (Europay, MasterCard, VISA) standard for authenticating credit and debit card transactions allows the same keypair to be reused for encryption and signature for precisely these reasons [12].

However, there is the question of whether it is secure to use the same keypair in two (or more) different primitives. The formal study of the security of key reuse was initiated by Haber et al. [13] in 2001, and followed by [9, 10, 20, 35]. Paterson et al. [31] gave examples, where encryption and signature schemes are individually secure but become completely insecure when a key pair is shared between them. They concluded that such scheme must be designed specially, and they gave a general construction and a more efficient concrete construction based on pairings, where encryption and signature schemes share the same keypair. They also proposed a scheme implementing the functionality of signcryption, signature and encryption using a single keypair.

Signcryption can realize signature and encryption simultaneously with lower computational costs and communication overheads than the traditional sign-then-encrypt approach, since it was first introduced by zheng [43] in 1997, many signcryption schemes [8, 28, 34, 38] have been proposed. However, sometimes we need confidentiality and authenticity simultaneously, and sometimes we just need them separately. To achieve this special requirement, we can naively use three different schemes: an encryption scheme, a signature scheme, and a signcryption scheme. Nevertheless, the naive approach need three keypairs, thus increases the burdens of the key management. In order to realize signcryption, signature, and encryption functions by using one keypair and one algorithm, so as to save storage spaces and simplify key management, Han et al. [17] in 2006 introduced a new concept of generalized signcryption (GSC). GSC scheme can produce the specific outputs according to the inputs of identities of the sender and the receiver adaptively, that is, if the input of the sender is vacant, it becomes an encryption scheme, if the input of the receiver is vacant, it becomes a signature scheme, if the inputs of the sender and the receiver are not vacant, it becomes a signcryption scheme, if the inputs of the sender and the receiver are all vacant, it takes no secure policy. Its main merit is the storage requirements for three schemes (signcryption, encryption and signature) and three key pairs can be reduced to one scheme and one key pair. Thus, it can realize using one keypair and one algorithm in three different cryptographic primitives. It is very suitable for storage-constrained environments, like the embedded systems, smart cards and wireless sensor networks.

Based on ECDSA [1] Han et al. [17] first proposed an efficient GSC scheme. Wang et al. [36] gave the first se-

curity model and revised Han et al.'s [17] scheme. In 2008, Lal et al. [25] gave the first identity-based generalized signcryption (ID-GSC) scheme and a security model of ID-GSC. In 2010, Yu et al. [40] pointed out Lal et al.'s [25] security model is not complete, and they improved it and proposed a new scheme which is secure in this model. Later, Kushwah et al. [22] simplified Yu et al.'s [40] security model and proposed another efficient ID-GSC scheme. Moreover, a lot of other GSC schemes have also been given out, including PKI-based (public key infrastructure) schemes [14, 16, 42], identity-based schemes [23, 26], certificateless schemes [19, 24, 30, 46], multi-PKG (private key generator) scheme [18, 45] and schemes in the standard model [18, 30, 37].

However all of the above mentioned schemes are suitable for one receiver scenario. Baudron et al. [2] and Bellare et al. [4] independently formalized the concept of multi-receiver public key encryption. Their main result is that the security of public key encryption in the single-receiver setting implies the security in the multi-receiver setting. Hence, one can construct a semantically secure multi-receiver public key encryption scheme by simply encrypting a message n times, obviously it is inefficient. Later, a novel technique called randomness reuse [21] was presented to enhance the efficiency. Randomness reuse is a novel technique to improve the efficiency of a multi-receiver encryption scheme, but not all randomness reuse-based multi-receiver encryption schemes are secure. Bellare et al. [3, 5] proved that if the underlying basic scheme is reproducible and semantically secure, then the corresponding randomness reuse-based multi-receiver encryption scheme is semantically secure too. Randomness reuse technique is also introduced to signcryption [11] and generalized signcryption [39] scenarios. Han et al. [15] proved if the underlying basic GSC scheme is reproducible and semantically secure, then the corresponding randomness reuse-based multi-receiver GSC scheme is semantically secure too.

In multi-receiver GSC setting, Han [14] first proposed a multi-receiver GSC scheme, but his scheme is a trivial n-receiver scheme that runs GSC repeatedly n times, which obviously is very inefficient. In 2008, Yang et al. [39] proposed a multi-receiver GSC scheme which used the technique of randomness reuse, but they did not give the security proof of their scheme. In 2009, Han et al. [15] proposed a multi-receiver GSC scheme, their scheme is very efficient and they applied it for secure multicast in wireless network. In 2014, Zhou [44] proposed the first time an identity-based multi-receiver GSC scheme which also used the technique of randomness reuse.

In this paper, we will show that Han et al.'s [15] basic GSC scheme and multi-receiver GSC scheme are insecure, their basic GSC scheme is not IND-CCA2 [32] secure in the pure encryption mode, and thus their multi-receiver GSC scheme is not IND-CCA2 secure in the pure encryption mode and hybrid encryption mode. Then we give an improvement of their scheme, interestingly, the improved scheme is more secure than the original one while still maintaining its efficiency. The confidentiality and existential unforgeability of the improved scheme can be proved under the CDH assumption. Compared with other multi-receiver signcryption schemes, our improved scheme enjoys shorter ciphertext length and less operation costs like the original scheme.

The rest of the paper proceeds as follows. In the next section, some preliminaries are listed. In Section 3, the definition and the security model of multi-receiver GSC schemes are given. In Section 4, we start with the description of Han et al.'s scheme, and give an attack on the scheme. In Section 5, we give an improvement of their scheme, and the security and performance analysis of the improved scheme. We conclude the paper in Section 6.

2 Preliminaries

Definition 1 (Bilinear pairings). *Let $k \in N$ be a security parameter and q be a k bits prime. We consider groups $(G_1, +)$ and $(G_2, \times)$ of the same prime order q . A bilinear map $e : G_1 \times G_1 \rightarrow G_2$ satisfies the following properties:*

1) *Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$, $a, b \in Z_q$.*

2) *Non-degeneracy: There exists $P, Q \in G_1$, such that $e(P, Q) \neq 1_{G_2}$.*

3) *Computability: There exists an efficient algorithm to compute $e(P, Q)$, for all $P, Q \in G_1$.*

Definition 2 (The Computational Diffie-Hellman problem). *The Computational Diffie-Hellman problem (CDH) in G_1 is to compute abP from $< P, aP, bP >$ for unknown randomly chosen $a, b \in Z_q$.*

The advantage of any probabilistic polynomial time algorithm G in solving CDH problem in G_1 is defined to be: $ADV_G^{CDH} = Pr[G(P, aP, bP) = abP : a, b \in Z_q^*]$.

CDH assumption: For every probabilistic polynomial time algorithm G, ADV_G^{CDH} is negligible.

3 Multi-receiver GSC Scheme and Its Security Notions

3.1 Syntax

A multi-receiver GSC scheme consists of the following three algorithms.

1) Setup algorithm:

   Given a secure parameter k, it generates the system public parameters. $(SK_X, PK_X) \leftarrow Gen(X, 1^k)$ is a key generation algorithm and produces the private key SK_X and the public key PK_X for the user X.

2) Generalized signcryption algorithm:

$\sigma \leftarrow GSC(M, SK_S, PK_{R_1}, PK_{R_2}, ..., PK_{R_n})$ is a probabilistic algorithm, and takes the private key SK_S of the sender S, the public keys $PK_{R_i}, (i = 1, ..., n)$ of the receivers and messages $M = m_i, (i = 1, ..., n)$ to return a ciphertext σ. There are 5 scenarios in this algorithm:

a. Pure signcryption mode:
If the sender and all of the receivers are determined, it runs in this mode, the ciphertext is
$\sigma \leftarrow GSC(M, SK_S, PK_{R_1}, PK_{R_2}, ..., PK_{R_n})$
$= signcrypt(M, SK_S, PK_{R_1}, PK_{R_2}, ..., PK_{R_n})$.

b. Pure signature mode:
If all of the receivers are vacant and the sender is determined, it runs in this mode, the ciphertext is
$\sigma \leftarrow GSC(M, SK_S, \phi_{R_1}, \phi_{R_2}, ..., \phi_{R_n})$
$= sign(M, SK_S)$. Here, ϕ means the user is vacant.

c. Pure encryption mode:
If the sender is vacant and all of the receivers are determined, it runs in this mode, the ciphertext is
$\sigma \leftarrow GSC(M, \phi_S, PK_{R_1}, PK_{R_2}, ..., PK_{R_n})$
$= encrypt(M, PK_{R_1}, PK_{R_2}, ..., PK_{R_n})$.

d. Hybrid signcryption mode:
If some of the receivers are vacant, and the rest of receivers and the sender are determined, it runs in this mode. For the determined receivers, the ciphertext σ is a signcryption ciphertext; for the vacant receivers, the ciphertext σ is a signature.

e. Hybrid encryption mode:
If some of the receivers and the sender are vacant, it runs in this mode. For the determined receivers, the ciphertext σ is an encryption ciphertext; for the vacant receivers, the ciphertext σ is a plaintext, it takes no secure policy.

3) De-generalized signcryption algorithm:

$m_i \cup \perp \leftarrow DGSC(\sigma_i, SK_{R_i}, PK_S)$ is a deterministic de-generalized signcryption algorithm and takes the public key PK_S of the sender S, the private key SK_{R_i} of the receiver R_i, and a ciphertext $\sigma_i \in \sigma(i = 1, ..., n)$, to return the message m_i or an invalid symbol $\perp$. There are 5 scenarios in this algorithm:

a. Pure signcryption mode: $DGSC(\sigma_i, SK_{R_i}, PK_S) = unsigncrypt(\sigma_i, SK_{R_i}, PK_S)$.

b. Pure signature mode: $DGSC(\sigma_i, \phi_{R_i}, PK_S) = verify(\sigma_i, PK_S)$.

c. Pure encryption mode: $DGSC(\sigma_i, SK_{R_i}, \phi_S) = decrypt(\sigma_i, SK_{R_i})$.

d. Hybrid signcryption mode:
For the determined receivers, $DGSC(\sigma_i, SK_{R_i}, PK_S) = unsigncrypt(\sigma_i, SK_{R_i}, PK_S)$; for the vacant receivers, $DGSC(\sigma_i, \phi_{R_i}, PK_S) = verify(\sigma_i, PK_S)$.

e. Hybrid encryption mode:
For the determined receivers, $DGSC(\sigma_i, SK_{R_i}, \phi_S) = decrypt(\sigma_i, SK_{R_i})$; for the vacant receivers, the ciphertext is the plaintext, it takes no secure policy.

For consistency, we require $DGSC(GSC(M, SK_S, PK_{R_1}, PK_{R_2}, ..., PK_{R_n}), SK_{R_i}, PK_S) = m_i$, for $i = 1, 2, ..., n, M = m_i$.

If all of the identities are vacant, it takes no secure policy. Above five modes are transparent to applications, namely, the algorithm can produce the specific outputs according to the inputs of identities of the sender and the receivers adaptively. Applications need not care about which mode should be taken.

3.2 Security Model of Multi-receiver GSC

The security notions for signcryption scheme are indistinguishability against adaptive chosen ciphertext attack (IND-SC-CCA2) and existential unforgeability against adaptive chosen message attack (EUF-SC-CMA). We modify these definitions to adapt for the multi-receiver GSC scheme. Namely, a multi-receiver GSC scheme should satisfy confidentiality (IND-MGSC-CCA2) and unforgeability (EUF-MGSC-CMA).

Definition 3. *A multi-receiver GSC scheme is said to be IND-MGSC-CCA2 secure if no probabilistic polynomial time adversary has a non-negligible advantage in the following game.*

1) *The challenger C runs Setup algorithm to generate the system public parameters and to generate multiple key pairs $(SK^*_{U_i}, PK^*_{U_i}), (i = 1, ..., n)$. $SK^*_{U_i}$ is kept secret while $PK^*_{U_i}$ is given to adversary A. These key pairs are the challenge key pairs. (Note: some of the key pairs can be null, it means the user is vacant. At least one key pair is not null.)*

2) ***Phase 1:*** *A makes polynomially bounded number of queries to the following oracles.*

a. *GSC Oracle: A produces messages $M = \{m_i, i = 1, \cdots, n\}$ and n arbitrary public keys $PK_{R_i}, (i = 1, ..., n)$ and requires the result of the operation $\sigma = GSC(M, SK^*_{U_j}, PK_{R_1}, ..., PK_{R_n})$ for an attacked user's private key $SK^*_{U_j}, (j \in [1, n])$. Challenger C runs GSC algorithm and returns the output σ to A.*

b. DGSC Oracle: A produces a ciphertext σ, an arbitrary public key PK_S of the sender and requires the result of $DGSC(\sigma, SK^*_{U_j}, PK_S)$ for the attacked users's private key $SK^*_{U_j}, (j \in [1,n])$. C runs DGSC algorithm and returns the output of DGSC to A.

These queries can be asked adaptively.

3) ***Challenge:*** *A produces two message vectors $M^*_0 = \{m^*_{0i}, i = 1, ..., n\}$, $M^*_1 = \{m^*_{1i}, i = 1, ..., n\}$, an arbitrary private key SK^*_S, B flips a coin $b \leftarrow \{0,1\}$ to compute a ciphertext $\sigma^* = GSC(M^*_b, SK^*_S, PK^*_{U_1}, ..., PK^*_{U_n})$ under the attacked users's public keys $PK^*_{U_j}, (j \in [1,n])$. B returns σ^* to A as a challenge.*

4) ***Phase 2:*** *A is allowed to make polynomially bounded number of new queries as in phase 1 with the restriction that A should not query the $DGSC(\sigma^*, SK^*_{U_j}, PK^*_S)$, $(j \in [1, n])$.*

5) ***Guess:*** *At the end of this game, A outputs a bit b_0. A wins the game if $b_0 = b$.*

The advantage of the adversary A is defined as follows: $Adv^{IND-MGSC-CCA2}(A) := 2Pr[b_0 = b] - 1$.

Note: In confidentiality game, it is only need to consider pure encryption mode, hybrid encryption mode, pure signcryption mode and hybrid signcryption mode with determined receivers. These modes share the same game. In the above challenge stage, the sender S can be vacant. In this case, algorithm runs in pure encryption mode or hybrid encryption mode, otherwise it runs in pure signcryption mode or hybrid signcryption mode. Because in the hybrid signcryption mode with vacant receivers, only signatures are made, it needs not to consider the IND-MGSC-CCA2 security.

Definition 4. *A multi-receiver GSC scheme is said to be EUF-MGSC-CMA secure if no probabilistic polynomial time adversary A has a non-negligible advantage in the following game.*

1) *The challenger C runs Setup algorithm to generate the system public parameters and to generate a key pair (SK^*_S, PK^*_S). SK^*_S is kept secret while PK^*_S is given to adversary A. The key pair can not be null and is considered as the challenge key pair. Because in the pure signature mode, pure signcryption mode or hybrid signcryption mode, the sender can not be vancant.*

2) ***Attack:*** *A can adaptively perform queries to the same oracles as those defined in Definition 3.*

3) ***Forgery:*** *At the end of the game, A produces a ciphertext σ^* and n arbitrary receivers's key pairs $(SK^*_{R_i}, PK^*_{R_i}), (i = 1, ..., n)$. A wins the game if the result of $DGSC(\sigma^*, PK^*_S, SK^*_{R_i}), (i \in [1,n])$ is a valid message m^*_i under the attacked users public key PK^*_S and the i-th receivers secret key $SK^*_{R_i}$, and σ^* is not the output of $GSC(M^*, SK^*_S, PK^*_{R_1}, ..., PK^*_{R_n})$, $M^* = \{m^*_1, m^*_2, ..., m^*_n\}$. A's advantage is its probability of victory.*

Note: In unforgeability game, it is only need to consider pure signature mode, pure signcryption mode and hybrid signcryption mode. These modes share the same game. In the above forgery stage, part or all of the receivers R^*_i can be vacant. In that case, algorithm runs in hybrid signcryption mode or pure signature mode, otherwise it runs in pure signcryption mode.

4 Han et al.'s Scheme and Its Security Analysis

According to the result of Bellare et al. [3, 5], if the underlying basic scheme is reproducible and semantically secure, then the corresponding randomness reuse-based multi-receiver encryption scheme is semantically secure too. Han et al. [15] extended the result to multi-receiver GSC setting, and proved that if the underlying basic GSC scheme is reproducible and semantically secure, then the corresponding randomness reuse-based multi-receiver GSC scheme is semantically secure too. So, Han et al. presented an underlying basic GSC scheme first, and then they proved the basic GSC scheme is reproducible and semantically secure, and concluded their multi-receiver GSC scheme is semantically secure.

4.1 Han-Gui's Basic GSC Scheme

The basic scheme is a GSC scheme suitable for one receiver and comes from the BLS signature [6]. The basic scheme is given as follows.

Setup: Let k be a secure parameter, q be a k bits prime, and G_1 be a bilinear group with order q. P is a generator of group G_1. Elements on G_1 have the length of l bits. $H_1 : \{0,1\}^z \times G_1 \rightarrow G_1$ and $H_2 : G_1^3 \rightarrow \{0,1\}^{z+l}$ are two hash functions, where z is the bit length of message m. In order to get adaptive outputs, they defined a special function $f(P)$. When $P = O$,$f(P) = 0$,else $f(P) = 1$, where $P \in G_1$ is a user's public key. $O \in G_1$ is the zero element.

Keygen: It takes the secure parameter k and users' identities to produce keys. For the sender S, his key pairs are $(x_s, Y_s) \leftarrow Gen(S, 1^k)$, where $x_s \in_R Z_q$ and $Y_s = x_s P \in G_1$. For the receiver R, his key pairs are $(x_R, Y_R) \leftarrow Gen(R, 1^k)$, where $x_R \in_R Z_q$ and $Y_R = x_R P \in G_1$. If $S \in \phi$ (an vacant user), $(0, O) \leftarrow Gen(S, 1^k)$. If $R \in \phi$, $(0, O) \leftarrow Gen(R, 1^k)$.

GSC: To signcrypt a z bits plaintext $m \in \{0,1\}^z$ to the intended receiver R, the sender S uses the following procedure.

1) Picks a random coin $r \in_R Z_q$ and computes $U = rP \in G_1$.
2) Computes $V = x_s H_1(m, rY_R) \in G_1$.
3) Computes $Z = (m||V) \oplus (H_2(U, Y_R, rY_R) f(Y_R)) \in \{0,1\}^{z+l}$.

The signcryption text is given by $\sigma = (U, Z) \in G_1 \times \{0,1\}^{z+l}$.

DGSC: When receiving $\sigma = (U, Z)$, the receiver R performs the steps below.

1) Computes $H_2(U, Y_R, x_R U) \in \{0,1\}^{z+l}$.
2) Computes $(m||V) = Z \oplus (H_2(U, Y_R, x_R U) f(Y_R))$.
3) If $V = O$, returns the message m, else computes $h = H_1(m, x_R U) \in G_1$ and then checks if $e(Y_s, h) = e(P, V)$. If this condition does not hold, rejects the ciphertext.

Pure signcryption mode: If the sender and the receiver are determined, it runs in this mode. Now, $x_s \neq 0$ and $f(Y_R) = 1$, the GSC and $DGSC$ algorithms are the same as above.

Pure encryption mode: If the sender is vacant and the receiver is determined, it runs in this mode. Now,$x_s = 0$ and $f(Y_R) = 1$, so, $V = x_s H_1(m, rY_R) = O$, $Z = (m||O) \oplus H_2(U, Y_R, rY_R)$, message m can be recovered by $(m||O) = Z \oplus H_2(U, Y_R, x_R U)$.

Pure signature mode: If the receiver is vacant and the sender is determined, it runs in this mode. Now, $x_s \neq 0$ and $f(Y_R) = 0$, so,

$$\begin{aligned} V &= x_s H_1(m, O), \\ Z &= (m||V) \oplus (H_2(U, Y_R, rY_R) f(Y_R)) \\ &= m||V. \end{aligned}$$

The signature can be verified by checking

$$e(Y_s, H_1(m, O)) = e(P, V).$$

If all of the identities are absent, it takes no secure policy. The three modes are transparent to applications, namely, the algorithm can produce the specific outputs according to the inputs of the identities of the sender and the receiver adaptively. Applications need not care about which mode should be taken.

4.2 Han et al.'s Multi-receiver GSC Scheme

A sender S sends z bits message vector $M = \{m_i | m_i \in \{0,1\}^z, i = 1, ..., n\}$ to intended receivers $R_i, (i = 1, ..., n)$, and then broadcasts the aggregated signcryption text. A receiver R_i gets his signcryption text and designcrypts it.

Setup: The same as above.

Keygen: It takes the secure parameter k and users' identities to produce keys. For the sender S, his key pairs are $(x_s, Y_s) \leftarrow Gen(S, 1^k)$, where $x_s \in_R Z_q$ and $Y_s = x_s P \in G_1$. For the receiver $R_i, (i = 1, ..., n)$, his key pairs are $(x_{R_i}, Y_{R_i}) \leftarrow Gen(R, 1^k)$, where $x_{R_i} \in_R Z_q$ and $Y_{R_i} = x_{R_i} P \in G_1$. If $S \in \phi$ (an vacant user), $(0, O) \leftarrow Gen(S, 1^k)$. If $R_i \in \phi$, $(0, O) \leftarrow Gen(R_i, 1^k)$.

GSC: To signcrypt message vector $M = \{m_i | m_i \in \{0,1\}^z, i = 1, \cdots, n\}$, S performs the following operations.

1) Picks a random coin $r \in_R Z_q$ and computes the commitment $U = rP \in G_1$.
2) For $i = 1, ..., n$
   a. Computes $V_i = x_s H_1(m_i, rY_{R_i}) \in G_1$.
   b. Computes $Z_i = (m_i||V_i) \oplus (H_2(U, Y_{R_i}, rY_{R_i}) f(Y_{R_i})) \in \{0,1\}^{z+l}$.

   EndFor
3) The ciphertext vector is given by $\sigma = (U, Z_1, ..., Z_n)$ which is sent to the group via a broadcast channel.

DGSC: When receiving σ, the receiver R_i gets his signcryption text $\sigma_i = (U, Z_i)$ and performs the following steps.

1) Computes $H_2(U, Y_{R_i}, x_{R_i} U)$.
2) Computes $(m_i||V_i) = Z_i \oplus (H_2(U, Y_{R_i}, x_{R_i} U) f(Y_{R_i}))$.
3) If $V_i = O$, returns the message m_i, else computes $h_i = H_1(m_i, x_{R_i} U) \in G_1$ and then checks if $e(Y_s, h_i) = e(P, V_i)$. If this condition does not hold, rejects the ciphertext.

Correctness: If $\sigma_i = (U, Z_i)$ is a valid signcryption text, it is easy to see that $x_{R_i} U = rY_{R_i} = x_{R_i} rP$ and $(m_i||V_i)$ is decrypted correctly. Thus, $e(P, V_i) = e(P, x_s h_i) = e(x_s P, h_i) = e(Y_s, h_i)$ holds.

Pure signcryption mode: If the sender and all of the receivers are determined, it runs in this mode. Now, $x_s \neq 0$ and $f(Y_{R_i}) = 1$, $(i = 1, 2..., n)$, the ciphertext vector $\sigma = (U, Z_1, ..., Z_n)$ is a signcryption ciphertext vector, the GSC and $DGSC$ algorithms are the same as above.

Pure encryption mode: If the sender is vacant and all of the receivers are determined, it runs in this mode. Now, $x_s = 0$ and $f(Y_{R_i}) = 1$, $(i = 1, 2..., n)$, so, $V_i = x_s H_1(m_i, rY_{R_i}) = O$, $Z_i = (m_i||O) \oplus H_2(U, Y_{R_i}, rY_{R_i})$, the ciphertext vector $\sigma = (U, Z_1, ..., Z_n)$ is a encryption ciphertext vector, message m_i can be recovered by $(m_i||O) = Z_i \oplus H_2(U, Y_{R_i}, x_{R_i} U)$.

Pure signature mode: If all of the receivers are vacant and the sender is determined, it runs in this mode. Now, $x_s \neq 0$ and $f(Y_{R_i}) = 0$, $(i = 1, 2..., n)$, so, $V_i = x_s H_1(m_i, O)$, $Z_i = (m_i||V_i) \oplus (H_2(U, Y_{R_i}, rY_{R_i})f(Y_{R_i})) = m_i||V_i$, the ciphertext vector $\sigma = (U, Z_1, ..., Z_n)$ is a signature vector, the signature can be verified by checking $e(Y_s, H_1(m_i, O)) = e(P, V_i)$.

Hybrid signcryption mode: If some of the receivers are vacant, and the rest of receivers and the sender are determined, the scheme runs in this mode. For the determined receivers, $x_s \neq 0$ and $f(Y_{R_i}) = 1$, the ciphertext vector $\sigma = (U, Z_i)$ is a signcryption ciphertext vector, and the procedure is the same as pure signcryption mode; for the vacant receivers, $x_s \neq 0$ and $f(Y_{R_i}) = 0$, the ciphertext vector $\sigma = (U, Z_i)$ is a signature vector, the procedure is the same as pure signature mode.

Hybrid encryption mode: If some of the receivers and the sender are vacant, it runs in this mode. For the determined receivers, $x_s = 0$ and $f(Y_{R_i}) = 1$, the ciphertext vector $\sigma = (U, Z_i)$ is a encryption ciphertext vector, and the procedure is the same as pure encryption mode; for the vacant receivers, $x_s = 0$ and $f(Y_{R_i}) = 0$, the ciphertext vector $\sigma = (U, Z_i)$ is a plaintext vector, it takes no secure policy.

The five modes are transparent to applications, namely, the algorithm can produce the specific outputs according to the inputs of identities of the sender and the receivers adaptively. Applications need not care about which mode should be taken.

4.3 An Attack on Han et al.'s Basic GSC Scheme Running in the Pure Encryption Mode

The security of Han et al.'s multi-receiver GSC scheme [15] relies on their basic GSC scheme. In the following, we will prove that Han et al.'s basic GSC scheme is not IND-CCA2 secure in the pure encryption mode, so their multi-receiver GSC scheme is insecure. Now we give an attack on the basic GSC scheme running in the pure encryption mode as follows.

Notice that in the pure encryption mode, $V = O$. Now assume that given the challenge receiver's public key Y_R^*, the adversary A chooses two equal length messages m_0^* and m_1^* and sends them to the challenger. The challenger then chooses a random $b \in \{0, 1\}$ and computes the challenge ciphertext of the message m_b^* as $\sigma^* = (U^*, Z^*)$ under the challenge public key Y_R^*. Upon receipt of the challenge ciphertext $\sigma^* = (U^*, Z^*)$, A chooses a random message $\overline{m}$, whose length is equal to that of m_0^*, and computes $\overline{Z} = Z^* \oplus (\overline{m}||O)$. Finally, the adversary A sends the ciphertext $\overline{\sigma} = (U^*, \overline{Z})$ to the challenger for decryption, obviously the challenger will return $(\overline{m} \oplus m_b^*)||O$ as the response, knowing the $\overline{m}$, A can get the m_b^*. Therefore, the basic GSC scheme is not IND-CCA2 secure in the pure encryption mode, and Han-Gui's multi-receiver GSC scheme is based on the basic GSC scheme so their multi-receiver GSC scheme is not IND-CCA2 secure in the pure encryption mode and hybrid encryption mode.

5 An Improved Multi-receiver GSC Scheme

5.1 An Improved Basic GSC Scheme

GSC: To signcrypt a z bits plaintext $m \in \{0, 1\}^z$ to the intended receiver R, the sender S uses the following procedure.

1) Computes $f(Y_s), f(Y_R)$.
2) Picks a random coin $r \in_R Z_q$ and computes $U = rP \in G_1$.
3) Computes $H = H_1(m, rY_R) \in G_1$,$V = x_s H \in G_1$.
4) If $f(Y_s) = 0$, Computes $Z = (m||H) \oplus (H_2(U, Y_R, rY_R)f(Y_R)) \in \{0, 1\}^{z+l}$, else computes $Z = (m||V) \oplus (H_2(U, Y_R, rY_R)f(Y_R)) \in \{0, 1\}^{z+l}$.

The signcryption text is given by $\sigma = (U, Z) \in G_1 \times \{0, 1\}^{z+l}$.

DGSC: When receiving $\sigma = (U, Z)$, the receiver R performs the steps below.

1) Computes $f(Y_s), f(Y_R)$.
2) If $f(Y_s) = 0$, Computes $(m||H) = Z \oplus (H_2(U, Y_R, x_R U)f(Y_R))$; else Computes $(m||V) = Z \oplus (H_2(U, Y_R, x_R U)f(Y_R))$.
3) Computes $h = H_1(m, x_R U) \in G_1$.
4) If $f(Y_s) = 0$, checks if $H = h$; if this condition does not hold, rejects the ciphertext; else returns m; else checks if $e(Y_s, h) = e(P, V)$; if this condition does not hold, rejects the ciphertext; else returns m.

5.2 An Improved Multi-receiver GSC Scheme

GSC: To signcrypt message vector $M = \{m_i | m_i \in \{0, 1\}^z, i = 1, ..., n\}$, S performs the following operations.

1) Computes $f(Y_s), f(Y_{R_i}), i = 1, ..., n$.
2) Picks a random coin $r \in_R Z_q$ and computes the commitment $U = rP \in G_1$.
3) For $i = 1, ..., n$
   a. Computes $H_i = H_1(m_i, rY_{R_i}) \in G_1$, $V_i = x_s H_i$.

b. If $f(Y_s) = 0$, Computes $Z_i = (m_i||H_i) \oplus (H_2(U, Y_{R_i}, rY_{R_i})f(Y_{R_i})) \in \{0,1\}^{z+l}$, else computes $Z_i = (m_i||V_i) \oplus (H_2(U, Y_{R_i}, rY_{R_i})f(Y_{R_i})) \in \{0,1\}^{z+l}$;

EndFor

4) The ciphertext vector is given by $\sigma = (U, Z_1, ..., Z_n)$ which is sent to the group via a broadcast channel.

DGSC: When receiving σ, the receiver R_i gets his signcryption text $\sigma_i = (U, Z_i)$ and performs the following steps.

1) Computes $f(Y_s), f(Y_{R_i}), i \in [1, n]$.
2) If $f(Y_s) = 0$, Computes $(m_i||H_i) = Z_i \oplus (H_2(U, Y_{R_i}, x_{R_i}U)\ f(Y_{R_i}))$, else computes $(m_i||V_i) = Z_i \oplus (H_2(U, Y_{R_i}, x_{R_i}U)f(Y_{R_i}))$.
3) Computes $h_i = H_1(m_i, x_{R_i}U) \in G_1$.
4) If $f(Y_s) = 0$, checks if $H_i = h_i$; if this condition does not hold, rejects the ciphertext; else returns m_i; else checks if $e(Y_s, h_i) = e(P, V_i)$, if this condition does not hold, rejects the ciphertext; else returns m_i.

5.3 Security Analysis

We have showed Han et al.'s underlying basic GSC scheme is not semantically secure in Paragraph 4.3, and Han-Gui's multi-receiver GSC scheme is based on the basic GSC scheme so their multi-receiver GSC scheme is not semantically secure either. The essence of their basic GSC scheme being insecure is as follows. Note that in the pure signcryption mode of their basic GSC scheme, the V part is not null, which intuitively makes it achives IND-CCA2 secure in the confidentiality game, and in the pure encryption mode, the V part is null, so the attacker can modify the challenge ciphertext to dechipher oracle to get the plaintext. In the improved basic GSC scheme, we use the H part to replace V part to concatenate message m in the pure encryption mode, which intuitively can make it achives IND-CCA2 secure.

About the security of the improved multi-receiver GSC scheme, we have the following two theorems. In proving the following two theorems, we reference the method adopted by [11, 27, 33, 41]. Their schemes all use the randomness reuse technique and they directly demonstrate their multi-receiver signcryption schemes rather than rely on a basic signcryption scheme. This method is different from Han et al.'s [15].

Theorem 1. *In the random oracle model with secure parameter k, if an adversary A has non-negligible advantage ε against the IND-MGSC-CCA2 security of the improved multi-receiver GSC scheme running in the pure encryption mode, hybrid encryption mode, pure signcryption mode or hybrid signcryption mode with determined receivers, A runs in time t and performs q_{GSC} GSC queries, q_{DGSC} DGSC queries and q_{H_i} queries to oracles $H_i, (i = 1, 2)$, then there exists an algorithm B that solves the CDH problem in G_1 with a probability $\varepsilon' \geq \varepsilon - (\frac{q_{H_2} q_{DGSC}}{2^k})$ in a time $t' \leq t + (2q_{DGSC} + 2q_{H_2})t_e$, where t_e denotes the time required for one pairing computation.*

Proof. We show how to build an algorithm B that solves the CDH problem by running the adversary A as a subroutine. On input (P, aP, b_iP), $(i = 1, 2, ..., n)$,the value of a, b_i is unknown, B's goal is to compute one of the ab_iP, $(i = 1, 2, ..., n)$. B sets $Y^*_{R_i} = b_iP$ as the challenge public keys, and gives these public keys to adversary A. Here some of the key pairs can be null, namely, $b_i = 0$, $Y^*_{R_i} = O$ for some i, it means the user is vacant. At least one key pair is not null.

Phase 1: A performs a first series of queries of the following kinds that are handled by B as explained below:

Simulator: H_1, H_2
B maintains lists L_1, L_2, which keep track of the answers given to oracle queries on H_1, H_2. Upon a query on H_i, B first scans in the list L_i to check whether H_i is already defined for that input. If it is, the previously defined value is returned. Otherwise, B picks a random element $h_i \in Z^*_q$, returns h_iP to A and stores h_i, h_iP and the user's query input in L_i.

Simulator: $GSC(M, x_S, Y_{U_1}, Y_{U_2}, ..., Y_{U_n})$
A produces a message vector $M = \{m_i, i = 1, ..., n\}$ and n arbitrary public keys $Y_{U_i}, (i = 1, ..., n)$ and requires the result of the operation $\sigma = GSC(M, x_S, Y_{U_1}, ..., Y_{U_n})$.

1) If the public key of the sender Y_S is not one of the target public keys $b_iP, i \in [1, n]$, B just runs GSC algorithm as normal because B knows the private key of the sender.
2) If the public key of the sender Y_S is one of the target public keys $b_iP, i \in [1, n]$, then B proceeds as follows:
   a. Computes $f(Y_S), f(Y_{U_j}), j = 1, ..., n$.
   b. Chooses $r \in_R Z_q$, computes $U = rP$.
   c. Queries H_1 oracle with the inputs of (m_j, rY_{U_j}) to get h_{1j}, then computes $V_j = h_{1j}b_iP$, for $j = 1, ..., n$.
   d. Queries H_2 oracle with the inputs of (U, Y_{U_j}, rY_{U_j}) to get h_{2j}.
   e. If $f(Y_S) = 0$, Computes $Z_j = (m_j\ ||\ h_{1j}P) \oplus (h_{2j}f(Y_{U_j}))$, for $j = 1, ..., n$; else $Z_j = (m_j||V_j) \oplus (h_{2j}f(Y_{U_j}))$, for $j = 1, ..., n$.
   f. The ciphertext vector is $\sigma = (U, Z_1, \cdots, Z_n)$, which is returned to A.

Simulator: $DGSC(\sigma, x_{R_i}, Y_S)$
A produces a ciphertext $\sigma = (U, Z_1, ..., Z_n)$, an

arbitrary public key Y_S of the sender and requires the result of $DGSC(\sigma, x_{R_i}, Y_S)$ for $i \in [1, n]$.

1) If the public key of the receiver Y_{R_i} is not the target public key $b_iP, i \in [1, n]$, B just runs the $DGSC$ algorithm as normal because B knows the private key of the receiver.
2) If the public key of the receiver Y_{R_i} is the target public key $b_iP, i \in [1, n]$, then B proceeds as follows:
   a. Computes $f(Y_S), f(Y_{R_i}), i \in [1, n]$.
   b. If $f(Y_S) = 0$, B iterates in L_2 for each item h_2, computes $(m_i||H_i) = Z_i \oplus (h_2 f(Y_{R_i}))$, then checks if m_i is in L_1; if not, moves to the next item of L_2 and begins again, else retrieves $h_{1i}P$, and checks if $H_i = h_{1i}P$; if not, move to the next item of L_2 and begins again, else returns m_i and stop. If B goes through L_2, no m_i returns, then B returns an invalid symbol $\perp$.
   c. If $f(Y_S) = 1$, B iterates in L_2 for each item h_2, computes $(m_i||V_i) = Z_i \oplus (h_2 f(Y_{R_i}))$, then checks if m_i is in L_1; if not, moves to the next item of L_2 and begins again, else retrieves $h_{1i}P$, and checks if $e(Y_s, h_{1i}P) = e(P, V_i)$; if not, move to the next item of L_2 and begins again, else returns m_i and stop. If B goes through L_2, no m_i returns, then B returns an invalid symbol $\perp$.

Challenge: A produces two message vectors $M_0 = \{m_{0i}, i = 1, ..., n\}$, $M_1 = \{m_{1i}, i = 1, ..., n\}$, an arbitrary private key x_S^*, and requires the GSC ciphertext on one of the two message vectors with the receiver public keys are the challenge public keys b_iP, $i = 1, ..., n$. B then sets $U^* = aP$, chooses $\{Z_1^*, Z_2^*, ..., Z_n^*\} \in_R \{0, 1\}^{z+l}$ and sends the challenge ciphertext $\sigma^* = (U^*, Z_1^*, ..., Z_n^*)$ to A.

Phase 2: A performs new queries as in phase 1 with the restriction that A should not query the $DGSC(\sigma^*, x_{R_i}^*, Y_S^*)$.

At the end of the game, A returns a guess. A cannot realize that σ^* is not a valid ciphertext unless A asks for one of the hash value $H_2(U^*, Y_{R_i}^*, aY_{R_i}^*) = H_2(aP, b_iP, ab_iP)$, $(i = 1, 2, ..., n)$, for which $b_i \neq 0$. B ignores A's answer and looks into the list L_2 for tuples of the form $(aP, b_iP, X, .)$. For each of them, B checks whether $e(P, X) = e(aP, b_iP)$, if this relation holds, B stops and outputs X as the solution of the CDH problem. If no tuple of this kind satisfies the above equality, B stops and outputs failure.

Now, we assess the probability that the simulation is not perfect. The only case where it can happen is when a valid ciphertext is rejected in a $DGSC$ query. It is easy to see that for every item in L_2, there is exactly one item in L_1 providing a valid ciphertext. The probability to reject a valid ciphertext is thus not greater than $\frac{q_{H_2}}{2^k}$. Since A makes total q_{DGSC} queries during the attack, so we have $\varepsilon' \geq \varepsilon - (\frac{q_{H_2} q_{DGSC}}{2^k})$. Moreover, the bound on B's computation time derives from the fact that every $DGSC$ query requires two pairing evaluations while the extraction of the solution from L_2 implies to compute at most $2q_{H_2}$ pairings. □

Note: In the above challenge stage, the sender S can be vacant. In this case, algorithm runs in pure encryption mode or hybrid encryption mode, otherwise it runs in pure signcryption mode or hybrid signcryption mode, these modes share the same game except in the hybrid signcryption mode with vacant receivers. Because in the hybrid signcryption mode with vacant receivers, only signatures are made, it needs not to consider the IND-MGSC-CCA2 security.

Theorem 2. *In the random oracle model with secure parameter k, if there exists a forger F with non-negligible advantage ε against the EUF-MGSC-CMA security of the improved multi-receiver GSC scheme running in the pure signature mode, pure signcryption mode or hybrid signcryption mode, F runs in time t and performs q_{GSC} GSC queries, q_{DGSC} DGSC queries and q_{H_i} queries to oracles $H_i, (i = 1, 2)$, then there exists an algorithm B that solves the CDH problem in G_1 with a probability $\varepsilon' \geq \varepsilon - (\frac{q_{H_2} q_{DGSC} + 1}{2^k})$ in a time $t' \leq t + (2q_{DGSC})t_e$, where t_e denotes the time required for one pairing computation.*

Proof. We show how to build an algorithm B that solves the CDH problem by running the adversary F as a subroutine. On input (P, aP, bP), B's goal is to compute abP. B sets $Y_S^* = bP$ as the challenge public key, and gives the public key to adversary F. The value of b can not be zero, because in the pure signature mode, pure signcryption mode or hybrid signcryption mode, the sender can not be vancant.

Attack: F issues queries to the same oracles as those in the confidentiality game and all oracles are the same except oracle H_1.

Simulator: H_1

B maintains a list L_1, which keeps track of the answers given to oracle queries on H_1. Upon a query (m_i, P_{e_i}), B first scans in the list L_1 to check whether H_1 is already defined for that input. If it is, the previously defined value is returned. Otherwise, B picks a random element $h_{1i} \in Z_q^*$ and sets $H_{1i} = h_{1i}aP$, and stores $(m_i, P_{e_i}, h_{1i}, H_{1i})$ in L_1, output H_{1i} to adversary F.

Forgery: F eventually produces a ciphertext $\sigma^* = (U^*, Z_1^*, ..., Z_n^*)$ and n arbitrary receivers's key pairs

$(x^*_{R_i}, Y^*_{R_i}), (i = 1, ..., n)$, and the attacked user's public key Y^*_S. B runs $DGSC$ algorithm using the private keys $x^*_{R_i}$ and the attacked user's public key Y^*_S to get the m^*_i, V^*_i. If σ^* is valid, we have $e(Y^*_S, H_1(m^*_i, x^*_{R_i}U^*)) = e(P, V^*_i)$, then the list L_1 must contain an entry $(m^*_i,\ x^*_{R_i}U^*, h^*_{1i}, H^*_{1i})$ with overwhelming probability (otherwise, B stops and outputs failure). Then

$$\begin{aligned} V^*_i &= x^*_S H^*_{1i} \\ &= x^*_S(h^*_{1i}aP) \\ &= h^*_{1i}abP, \end{aligned}$$

and that $(h^*_{1i})^{-1}V^*_i$ is the solution of the CDH instance abP.

Now we assess B's probability of failure, F outputs a fake σ^* without asking the corresponding $H_1(m^*_i, x^*_{R_i}U^*, h^*_{1i}, H^*_{1i})$ query is at most $\frac{1}{2^k}$, The probability to reject a valid ciphertext is not greater than $\frac{q_{H_2}q_{DGSC}}{2^k}$. Finally, it comes that B's advantage is $\varepsilon' \geq \varepsilon - (\frac{q_{H_2}q_{DGSC}+1}{2^k})$. Moreover, the bound on B's computation time derives from the fact that every $DGSC$ query requires two pairing evaluations. □

Note: In the above forgery stage, part or all of the receivers R_i can be vacant. In that case, algorithm runs in hybrid signcryption mode or pure signature mode, otherwise it runs in pure signcryption mode, these modes share the same game.

5.4 Performance Analysis

Since computation time and ciphertext size are two important factors affecting the efficiency, we present the comparison with respect to them. It is obvious that our improved scheme does not add any extra computation costs and the ciphertext size is the same as the original one, meaning they have the same efficiency, but the original one is not secure while ours is. The authors of the original scheme compared their scheme with other multi-receiver signcryption schemes including Duan et al.'s multi-receiver signcryption [11] (denoted by DC), Yu et al.'s signcryption [41] (denoted by YYHZ), Li et al.'s identity-based broadcast signcryption [29] (denoted by LXH) and Boyens multipurpose identity-based signcryption [7] (denoted by Boyen). They considered the costly operations including pairing evaluation (Pairing), modular exponentiation (Exp), and modular inverse (Inv). Through the comparison, they concluded their scheme is the most efficient one. Therefore our improved scheme is the most efficient one too. Now, we give the comparison in Table 1, which shows that the computation time and ciphertext size of our improved scheme are both the shortest like the original scheme's.

6 Conclusions

Generalized signcryption scheme can adaptively work as an encryption scheme, a signature scheme or a signcryption scheme with only one algorithm and one key pair, thus it can realize using one keypair and one algorithm in three different cryptographic primitives. It is very suitable for storage-constrained environments. By using the randomness reuse technology, Han et al. proposed a multi-receiver GSC scheme, and used it for secure multicast in wireless network. Its main merits are to reduce overheads efficiently and avoid rekeying when membership changes. In this paper, we show that Han et al.'s multi-receiver GSC scheme is not IND-CCA2 secure in the pure encryption mode and the hybrid encryption mode, and an adversary can modify the challenge ciphertext and then can get the plaintext. To remedy this security flaw, we give an improvement of the scheme. Interestingly, the improved scheme is more secure than the original one while still maintaining its efficiency. Due to the computation of the pairing still being time-consuming, it is expected pairing-free multi-receiver GSC schemes are to be proposed in the future.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No.61462048, and the key program of Jiujiang University under Grant No. 2013ZD02. The author gratefully acknowledges the anonymous reviewers for their valuable comments.

References

[1] X9.62 ANSI, "The elliptic curve digital signature algorithm (ECDSA)," 1999. (http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf)

[2] O. Baudron, D. Pointcheval, and J. Stern, "Extended notions of security for multicast public key cryptosystems," in *Proceedings of ICALP'00*, pp. 499–511, 2000.

[3] M. Bellare, A. Boldyreva, K. Kurosawa, and J. Staddon, "Multi-recipient encryption schemes: how to save on bandwidth and computation without sacrificing security," *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 3927–3943, 2007.

[4] M. Bellare, A. Boldyreva, and S. Micali, "Public-key encryption in a multi-user setting: Security proofs and improvements," in *Proceedings of Eurocrypt'00*, pp. 259–274, 2000.

[5] M. Bellare, A. Boldyreva, and J. Staddon, "Randomness re-use in multi-recipient encryption scheme," in *Proceedings of Public Key Cryptography*, pp. 85–99, 2003.

[6] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proceedings of Asiacrypt'01*, pp. 514–532, 2001.

Table 1: Overheads of multi-receiver signcryption schemes

Schemes	Communication overheads	Computational overheads					
		Pairing		Exp		Inv	
		SC	DSC	SC	DSC	SC	DSC
DC	$(n+3)\|G\|+\|m\|+\|ID\|$	1	4n	n+5	n	0	2n
YYHZ	$(n+3)\|G\|+\|m\|+\|ID\|$	1	3n	n+5	n	0	n
LXH	$(n+2)\|G\|+\|m\|+\|ID\|$	1	3n	n+3	2n	0	0
Boyen	$2n\|G\|+\|m\|+\|ID\|$	n	4n	2n+2	2n	0	n
Original Scheme	$(n+1)\|G\|+\|m\|$	0	2n	n+1	n	0	0
Our Scheme	$(n+1)\|G\|+\|m\|$	0	2n	n+1	n	0	0

[7] X. Boyen, "Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography," in *Proceedings of Crypto'03*, pp. 383–399, 2003.

[8] H. Y. Chen, Y. Li, and J. P. Ren, "A practical identity-based signcryption scheme," *International Journal of Network Security*, vol. 15, no. 6, pp. 484–489, 2013.

[9] J. S. Coron, M. Joye, D. Naccache, and P. Paillier, "Universal padding schemes for rsa," in *Proceedings of Crypto'02*, pp. 226–241, 2002.

[10] J. P. Degabriele, A. Lehmann, and K. G. Paterson, "On the joint security of encryption and signature in emv," in *Proceedings of CT-RSA'12*, pp. 116–135, 2012.

[11] S. S. Duan and Z. F. Cao, "Efficient and provably secure multi-receiver identity-based signcryption," in *Proceedings of ACISP'06*, pp. 195–206, 2006.

[12] Corporation EMV, "EMV Book 3 - Application specification - Ver. 4.2," 2008. (http://www.emvco.com/)

[13] S. Haber and B. Pinkas, "Securely combining public-key cryptosystems," in *ACM Conference on Computer and Communications Security*, pp. 215–224, 2001.

[14] Y. L. Han, "Generalization of signcryption for resources-constrained environments," *Wireless Communications and Mobile Computing*, vol. 7, no. 7, pp. 919–931, 2007.

[15] Y. L. Han and X. L. Gui, "Adaptive secure multicast in wireless networks," *International Journal of Communication Systems*, vol. 22, no. 9, pp. 1213–1239, 2009.

[16] Y. L. Han and X. L. Gui, "Bpgsc: Bilinear paring based generalized signcryption scheme," in *Eighth International Conference on Grid and Cooperative Computing*, pp. 76–82, 2009.

[17] Y. L. Han and X. Y. Yang, "Ecgsc: Elliptic curve based generalized signcryption," in *The 3rd International Conference on Ubiquitous Intelligence and Computing (UIC'06)*, pp. 956–965, 2006.

[18] H. F. Ji, W. B. Han, and L. D. Liu, "Identity based generalized signcryption scheme for multiple pkgs in standard model," *Journal of Electronics and Information Technology (in Chinese)*, vol. 33, no. 5, pp. 1204–1210, 2011.

[19] H. F. Ji, W. B. Han, and L. Zhao, "Certificateless generalized signcryption," *Cryptology ePrint Archive*, 2010. (http://eprint.iacr.org/2010/204)

[20] Y. C. Komano and K. Ohta, "Efficient universal padding techniques for multiplicative trapdoor one-way permutation," in *Proceedings of CRYPTO'03*, pp. 366–382, 2003.

[21] K. Kurosawa, "Multi-recipient public-key encryption with shortened ciphertext," in *Proceedings of Public Key Cryptography*, pp. 48–63, 2002.

[22] P. Kushwah and S. Lal, "Efficient generalized signcryption schemes," *Cryptology ePrint Archive*, 2010. (http://eprint.iacr.org/2010/346)

[23] P. Kushwah and S. Lal, "An efficient identity based generalized signcryption scheme," *Theoretical Computer Science*, vol. 412, no. 45, pp. 6382–6389, 2011.

[24] P. Kushwah and S. Lal, "Provable secure certificateless generalized signcryption scheme," *International Journal of Computer Technology and Applications*, vol. 3, no. 3, pp. 925–939, 2012.

[25] S. Lai and P. Kushwah, "Id-based generalized signcryption," *Cryptology ePrint Archive*, 2008. (http://eprint.iacr.org/2008/084)

[26] S. Lai and P. Kushwah, "Generalization of barreto et al id based signcryption scheme," *Cryptology ePrint Archive*, 2009. (http://eprint.iacr.org/2009/193)

[27] F. G. Li, Y. P. Hu, and S. G. Liu, "Efficient and provably secure multi-recipient signcryption from bilinear pairings," *Wuhan University Journal of Nature Sciences*, vol. 12, no. 1, pp. 17–20, 2007.

[28] F. G. Li, X. J. Xin, and Y. P. Hu, "ID-based signcryption scheme with (t, n) shared unsigncryption," *International Journal of Network Security*, vol. 3, no. 2, pp. 155–159, 2006.

[29] F. G. Li, X. J. Xin, and Y. P. Hu, "Indentity-based broadcast signcryption," *Computer Standards and Interfaces*, vol. 30, no. 2, pp. 89–94, 2008.

[30] L. D. Liu, H. F. Ji, W. B. Han, and L. Zhao, "Certificateless generalized signcryption scheme without random oracles," *Journal of Software (in Chinese)*, vol. 23, no. 2, pp. 394–410, 2012.

[31] K. G. Paterson, J. C. N. Schuldt, M. Stam, and S. Thomson, "On the joint security of encryption and signature, revisited," in *Proceedings of Asiacrypt'11*, pp. 161–178, 2011.

[32] C. Racko and D. Simon, "Non-interactive zero knowledge proof of knowledge and chosen ciphertext attacks," in *Proceedings of Crypto'91*, pp. 433–444, 1991.

[33] S. S. D. Selvi, S. S. Vivek, R. Gopalakrishnan, N. N. Karuturi, and P. Rangan, "On the provable security of multi-receiver signcryption schemes," *Cryptology ePrint Archive*, 2008. (http://eprint.iacr.org/2008/238)

[34] M. Toorani and A. A. B. Shirazi, "Cryptanalysis of an elliptic curve-based signcryption scheme," *International Journal of Network Security*, vol. 10, no. 1, pp. 51–56, 2010.

[35] M. I. G. Vasco, F. Hess, and R. Steinwandt, "Combined (identity-based) public key schemes," *Cryptology ePrint Archive*, 2008. (http://eprint.iacr.org/2008/466)

[36] X. A. Wang, X. Y. Yang, and Y. L. Han, "Provable secure generalized signcryption," *Cryptology ePrint Archive*, 2007. (http://eprint.iacr.org/2007/173)

[37] G. Y. Wei, J. Shao, Y. Xiang, P. P. Zhu, and R. X. Lu, "Obtain confidentiality or/and authenticity in big data by ID-based generalized signcryption," *Information Sciences, DOI: 10.1016/j.ins.2014.05.034*, 2014.

[38] H. Xiong, J. B. Hu, and Z. Chen, "Security flaw of an ECC-based signcryption scheme with anonymity," *International Journal of Network Security*, vol. 15, no. 4, pp. 317–320, 2013.

[39] X. Y. Yang, M. T. Li, L. X. Wei, and Y. L. Han, "New ecdsa-verifiable multi-receiver generalization signcryption," in *The 10th IEEE International Conference on High Performance Computing and Communications*, pp. 1042–1047, 2008.

[40] G. Yu, X. X. Ma, Y. Shen, and W. B. Han, "Provable secure identity based generalized signcryption scheme," *Theoretical Computer Science*, vol. 411, no. 40-42, pp. 3614–3624, 2010.

[41] Y. Yu, B. Yang, X. Y. Huang, and M. W. Zhang, "Efficient identity-based signcryption scheme for multiple receivers," in *Proceedings of ATC'07*, pp. 13–21, 2007.

[42] C. R. Zhang and Y. Q. Zhang, "Secure and efficient generalized signcryption scheme based on a short ECDSA," in *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'10)*, pp. 466–469, 2010.

[43] Y. L. Zheng, "Digital signcryption or how to achieve cost (signature & encryption)<<cost (signature)+ cost (encryption)," in *Proceedings of Crypto'97*, LNCS 1294, pp. 165–179, 1997.

[44] C. X. Zhou, "Provably secure and efficient multi-receiver identity-based generalized signcryption scheme," in *2014 Ninth Asia Joint Conference on Information Security*, pp. 82–89, 2014.

[45] C. X. Zhou and Y. L. Han, "Identity-based multi-pkg generalized signcryption scheme," *Journal of Chinese Computer Systems (in Chinese)*, vol. 34, no. 7, pp. 1631–1636, 2013.

[46] C. X. Zhou, W. Zhou, and X. W. Dong, "Provable certificateless generalized signcryption scheme," *Designs Codes and Cryptography*, vol. 71, no. 2, pp. 331–346, 2014.

Cai-Xue Zhou received the B.S. in Computer Science Department from Fudan University in 1988, Shanghai, China and the M.S. in Space College of Beijing University of Aeronautics and Astronautics in 1991, Beijing, China. He is an Associate Professor in the School of Information Science and Technology, University of Jiujiang, Jiujiang, China since 2007. He is a member of the CCF (China Computer Federation), and a member of CACR(Chinese Association for Cryptologic Research). His research interests include applied cryptography, security of computer networks .

The Optimization Research of the Multimedia Packets Processing Method in NIDS with 0/1 Knapsack Problem

Xu Zhao
Department of Computer Science, Xian Polytechnic University, Xian 710048, China
(Email: 37274679@qq.com)
(Received Jan. 30, 2015; revised and accepted Mar. 26 & Apr. 6, 2015)

Abstract

There always exists a high packet loss rate in the network intrusion detection system, especially when the network traffic is high. The author raised the method of Multimedia Packets Processing to solve this problem and received good results. In this paper, on the basis of these research results, 0/1 knapsack problem has been used to optimize multimedia packets processing method. Thus, by using this modified method, an optimum solution can be found to select the highest risk of multimedia packets sequence in each time unit. Network intrusion detection system can make limited processing capability focus on the even riskier multimedia packets by using this improved method. Experiments have shown that the method can effectively improve the detection rate of multimedia packets with dangerous information. In addition, on one hand, the packet loss rate of network intrusion detection system decreases obviously, and on the other hand, its security is improved.

Keywords: 0/1 Knapsack problem, multimedia packets, network intrusion detection system

1 Introduction

Network intrusion detection system (NIDS) is the system of identifying and processing malicious behavior in network resources. As network speeds up, the requirements of NIDS's processing efficiency also increase. How to improve NIDS in the processing capacity per unit time becomes a research hotspot in the field [9, 10, 11, 15].

Sravani et al. [14] proposed the use of a classification algorithm for network traffic data classification to solve this problem. Jiang et al. [3] also proposed an improved ant colony clustering method for intrusion detection. This method has not only improved the detection rate, but also reduced the fault detection rate. However, this method has low detection efficiency for unknown attacks. Liu et al. [5] proposed an improved k-means clustering algorithm for NIDS, whose algorithm can effectively improve detection accuracy. But her algorithm has certain instability when the value of K is between 3 and 6. Arun et al. [1] discussed signature detection technique (SDT) used in network intrusion detection system. Meanwhile, Zhai [18] proposed a new intrusion detection algorithm based on the combination of K-prototypes and fuzzy evaluation. This algorithm not only improves the detection accuracy, but also reduces false detection rate compared to K-prototypes algorithm or fuzzy evaluation, but this algorithm can also mistake dubious data for normal data. Wu et al. [16] proposed an intrusion feature subset selection algorithm based on Particle Swarm Optimization. This algorithm can effectively remove redundant features, reduce the time of feature selection, ensure detection accuracy and improve detecting speed. Li et al. [4] proposed an intrusion detection model based on immune Agent and particle swarm optimization immune principle by means of combining mobile Agent and quantum-behaved particle swarm optimization. This system can improve the low detecting speed and high false positive rate of traditional NIDS. However, the detection rate of this method is not ideal for U2R and R2L. Shi et al. [12] proposed a simple and fast discretization algorithm based on information loss by fusing Rough Set theory with entropy theory. This algorithm is applied to different samples with the same attributes from KDDcup99 and intrusion detection systems. However, the proposed discretization algorithm is sensitive to the initial samples only for part of all condition attributes. In addition, some researchers [2, 6, 7, 13, 17] improved the efficiency of intrusion detection system by using improved pattern matching algorithm. For example, Lu et al. [7] present an improved multi-pattern matching algorithm which is based on deterministic finite-state automaton. However, this algorithm is only suitable for finding the small character sets pattern string in large character set text string.

Since the multimedia packets occupy a larger proportion in network flow, the method of particular processing on them can greatly improve the efficiency of NIDS. Among many studies on NIDS, starting from the study of multimedia data in network flow, I proposed the Multime-

dia Packets Processing Method [20] to solve this problem and gained very satisfying results. Based on these studies, the idea of optimization in 0/1 Knapsack problem is applied to the Multimedia Packets Processing Method to further enhance the efficiency of the method.

2 Multimedia Packets Processing Method

The conventional detection method of NIDS performs pattern matching on all packets by means of thousands of rules, which cannot distinguish between the multimedia packets and non-multimedia packets. Actually, this makes multimedia packets, which account for a larger proportion in network traffic and are relatively safe, consume a lot of system's resources.

To address this problem, the author have proposed and designed an identifying method and two separate processing methods [21] for multimedia packets, and on this basis, we have realized Multimedia Packets Processing Method [20]. Here are these two processing methods.

1) Releasing method: this method is comparatively simple. When multimedia packet is found in net flow, this method makes identified multimedia packet skip over the conventional process. Though this method is simple and efficient, its security is lower [21].

2) Corresponding media type detection method: This method is a safe and an efficient method. In order to achieve this method, a multimedia rule base is created. The multimedia rule base stores the rules which are specifically collected for multimedia packets. The corresponding media type detection method can be used to choose the corresponding multimedia rules according to the specific multimedia type that packet carries in order to pre-detect aggressive characteristics. If there is no problem, release it directly; if there exists a problem, put it into the conventional detection process. Because the number of multimedia rules is far less than that of rules for conventional detection process in NIDS, this method can significantly improve the detective efficiency for most of the safe multimedia packets. The safety of this method is also higher than that of the releasing method [21].

Compared with the conventional detection method (i.e. performing pattern match to all packets with thousands of rules) in the NIDS, both of the two processing methods can obviously reduce the packet loss rate (As shown in Table 1) and improve threshold value of dropping packets (As shown in Figure 1).

The relationship between the packet loss rate by using three different methods and bandwidth can be shown in Figure 1. According to this relationship, the author has proposed the Multimedia Packets Processing Method [20], whose basic idea is as follows:

Table 1: Comparison of three methods in the detection efficiency, packet loss rate and undetected rate

methods	detection efficiency	packet loss rate	undetected rate
Conventional detection method (default)	Low	High	Low
Releasing method	High	Low	High
Corresponding media type detection method	Medium	Medium	Medium

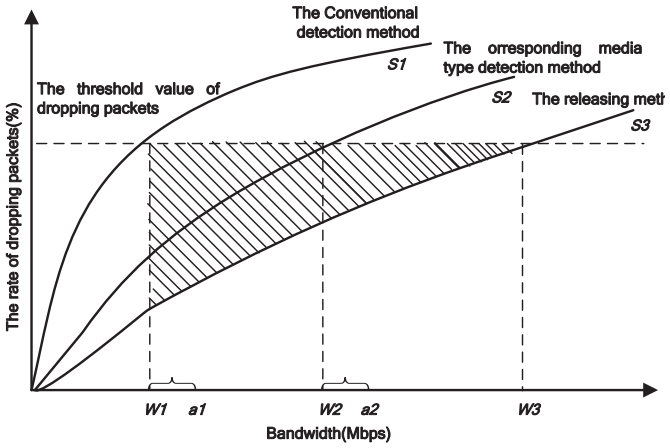


Figure 1: The threshold values of dropping packets by using three different methods

1) When system starts, it works with the conventional detection method and meanwhile statistics stable network traffic in a period of time. Then according to the speed of current network traffic (i.e. less than W1, or and W2, or between W2 and W3, or more than W3), the system can select the most appropriate method to work from the conventional detection methods, the releasing method and the corresponding media type detection method. If the speed of network traffic crosses from one interval to another interval, the system should reselect the method [20].

2) If the chosen method is the corresponding media type detection method and the speed of current network traffic is between W1 and W2 (within this interval the system has spare processing capacity), the system should process increasing number of multimedia packets by using the conventional detection method under the premise of no packets loss. If packet loss arises, the system should process decreasing number of multimedia packets [20].

3) If the chosen method is the releasing method and the speed of current network traffic is between W2 and W3 (within this interval, the system has spare processing capacity), the system should process increasing number of multimedia packets by using the corresponding media type detection method under the premise of no packets loss. If packet loss arises,

the system should process decreasing number of multimedia packets [20].

3 0/1 Knapsack Problem

0/1 Knapsack problem can be described as follows: there are n pieces of removable items (items number is from 1 to n) and a backpack whose maximum load is M. The weight of the item i is W_i, the benefits obtained for putting item i into the backpack are P_i. $M > 0, 0 \le i \le n, W_i > 0, P_i > 0$. An optimal loading method needs to be sought, which can get the largest benefits of the items in the backpack.

That is to say, under the condition: $\sum_{i=1}^{n} W_i X_i \le M$, $M > 0, W_i > 0, X_i = 0$ or $X_i = 1, 1 \le i \le n$, let objective function: $\sum_{i=1}^{n} P_i X_i$ ($X_i = 0$ or 1, $P_i > 0$) have maximum value. This has becomes 0/1 knapsack problem.

4 The Multimedia Packets Processing Method Based on 0/1 Knapsack Problem

Among many types of multimedia data in network flow, the risk of certain types of multimedia data is higher than that of others. Under the condition of the system processing ability being limited, these packets should be given priority treatment for processing. So in Steps (2) and (3) of the Multimedia Packets Processing Method, we hope that those more dangerous multimedia packets are processed preferentially by more rigorous methods(the Conventional detection method and the corresponding media type detection method) within the range of the maximum load in the system. Accordance with the idea of 0/1 knapsack problem, this optimization problem can be described as follows.

In the interval of W1∼W2 (or W2∼W3) in Figure 1, set the maximum load of the NIDS as M, the identified n multimedia packets ($X_{1,}X_{2,...,}X_n$) within each time slice needs to be picked out and be dealt with by the conventional detection method (if in the interval of W2∼W3, the corresponding media type detection method should be used), set the load to the system caused by multimedia packet X_i as W_i, set the risk factor of multimedia packet X_i as P_i. We need to find an optimal selection method, make the objective function:

$$\max \sum_{i=1}^{n} P_i X_i \quad X_i \in \{0,1\}, P_i > 0, 1 \le i \le n.$$

The constraint function:

$$\sum_{i=1}^{n} W_i X_i \le M, M > 0, W_I > 0, X_i \in \{0,1\}, 1 \le i \le n.$$

Wherein variable $X_i = 0$ means the multimedia packet X_i is not selected, $X_i = 1$ means it is selected.

4.1 The Determination Method of Three Parameters

In the process of implementation, M, W_i, P_i, these 3 parameters need to be determined, the specific methods are shown as follows.

1) M: M can be obtained by counting the stable simulated network traffic over a period of time under the condition of the same bandwidth.

2) W_i: because there are significant positive correlations between the time complexity of the pattern matching algorithm and length of the string to be matched, W_i is determined by the ratio of the actual length of the packet load and the total length.

3) P_i: the value of P_i depends on the risk factor of multimedia data type which the packets carry. If the packet contain media type such as x-javascript, octet-stream, html, jpeg, gif, x-shockwave-flash, etc. , then its P_i value will be higher.

4.2 Solving Process

The 0/1 Knapsack problem of n identified multimedia packets ($X_{1,}X_{2,\cdots,}X_n$) within a time slice is expressed as *Knap(1,n,M)*, the solution process is shown as following.

According to forward reasoning, if the decision order for X_i is $X = \{X_1, X_2, ..., X_n\}$, there exist two cases after making decisions on X_n:

1) $X_1 = 1$, meaning the multimedia packet X_1 is selected, then the sub-problem $Knap(2, n, M - W_1)$ should be solved.

2) $X_1 = 0$, meaning the multimedia packet X_1 is not selected, then the sub-problem $Knap(2, n, M)$ should be solved.

Let $f_j(M)$ is the value of the optimal solution to $Knap(j+I, n, M)$. That is, $f_j(M)$ is the value of the optimal solution when selectable multimedia packet is $j+1, j+2, \ldots, n, (1 \le j \le n)$ and maximum load of NIDS is M, then $f_0(M)$ can be expressed as:

$$f_0(M) = \max\{f_1(M), f_1(M - W_1) + P_1\}.$$

Because the value of X_i is 0 or 1, so formula can be derived as

$$f_j(x) = \max\{f_{j+1}(x), f_{j+1}(x - W_{j+1}) + P_{j+1}\}.$$

Set $X \ge 0$, then $f_0(x) = 0$ for $x < 0$, $f_0 = -\infty$. So, we can calculate the $f_0(M)$, which is the value of the optimal solution to Knap(1, n, M).

According to backward reasoning, if the decision order for X_i is $X = \{X_n, X_{n-1}, ..., X_1\}$, there exist two cases after making decisions on X_n:

1) $X_n = 1$, meaning the multimedia packet X_n is selected, then the sub-problem $Knap(1, n-1, M-W_n)$ should be solved;

2) $X_n = 0$, meaning the multimedia packet X_n is not selected, then the sub-problem $Knap(1, n-1, M)$ should be solved;

Let $f_i(M)$ is the value of the optimal solution to *Knap(i,n,M)*. That is, $f_i(M)$ is the value of the optimal solution when selectable multimedia packet is $n, \ldots, i+1, i, (1 \le i \le n)$ and maximum load of NIDS is M. Then $f_n(M)$ can be expressed as:

$$f_n = \max\{_{n-1}(M), f_n(M - W_n + P_n\}.$$

Recursion continues in turn until $f_1(M)$, there is:

$$f_1(M) = f_0(M), f_0(M - W_1) + P_1.$$

Set $X \ge 0$, then $f_0(x) = 0$ for $x < 0$, $f_0(x) = -\infty$. So, starting with the $f_0, f_n(M)$ can be calculated, which is the value of the optimal solution to *Knap (1, n, M)*.

Extending to general cases, there is:

$$f_i(x) = \max\{f_{i-1}(x), f_{i-1}(x - W_i) + P_i\}.$$

According to this formula, both the optimal solution and the optimal choice sequence of multimedia packets can be derived.

5 Experiment

5.1 Experimental Environment

The experiments reported here demonstrate a variety of changes in performance before and after optimization. Experimental environment consists of three computers (OS: WIN 7, CPU: Dual-core processor 3.00Ghz, Memory: 4GB DDR3).

In the experiment, the network traffic, which is captured before and sent by the first computer and also adapted as real background traffic, contains a large number of multimedia data packets. As the attacker, the second computer uses Lincoln Laboratory KDD CUP 99 data set and IDS Informer to generate attack traffic. Both mixed flows are sent to test NIDS installed on the third computer, as is shown in Figure 2.

In the attack traffic, including the four types of network attacks [8, 19]: DoS, R2L, U2R and PROBE, the types and quantities of attack category are shown in Table 2.

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

	DoS	R2L	U2R	PROBE	Total
number	229853	16137	228	4166	250384

Before testing, first of all, the value of P should be set for different media types according to the degree of risky information carried by packets (as shown in Table 3). For example, executable files can appear in multimedia file of octet-stream type, so the value of P of octet-stream type can be set higher. The table below shows the statistics of several common multimedia types and the value of P.

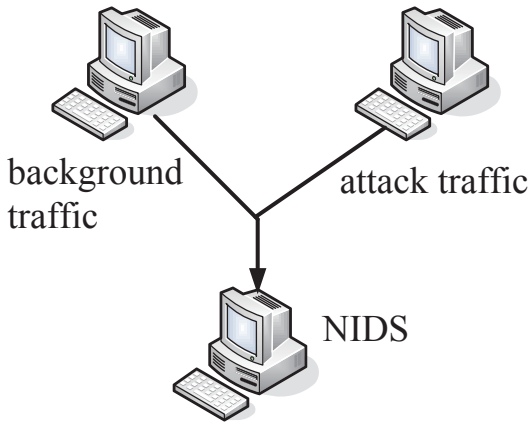


Figure 2: Experimental environment

5.2 Experimental Results and Analysis

Three experiments have provided herein, the first step of experiment is to compare differences between the risk factor which is determined by multimedia packets selected in different time slices before and after optimization (as shown in Figure 3).

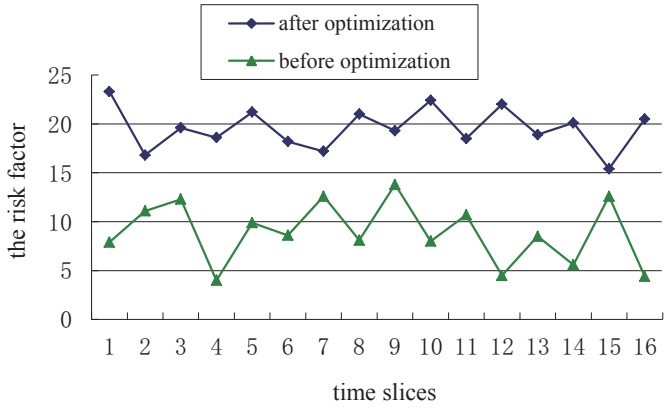


Figure 3: The differences between the risk factor in different time slices before and after optimization

As shown in the figure above, the degree of risk of multimedia packets selected in different time slices after optimization is significantly higher than that of before optimization. This is because multimedia packets are selected by chronological order regardless of the degree of risk of multimedia packets within a time slice before optimization. So the curve fluctuates significantly before optimization and shows its random, while the curve is relatively stable and higher after optimization. This can also be proved by the sample variance of results. According to the calculation formula of sample variance $S^2 = \frac{\sum_{i=1}^{n}(x_i - E(x))^2}{n-1}$, Sample variance is 6.4 before optimization, while it is 3.8 after optimization.

The second step of the experiment is to compare differences in the detection rates of different multimedia types of packets before and after optimization (as shown in Ta-

Table 3: the statistics of several common multimedia types and the value of P

multimedia type	number of packets	total amount of data(Byte)	Average packet length(Byte)	P
octet-stream	31	25916	836	2.4
x-javascript	35	135135	3861	2.2
html	43	2443690	56830	1.5
jpeg	58	2012716	34702	1.3
gif	156	365196	2341	1.3
x-shockwave-flash	27	20331	753	1.7
.....	...	...	...	...

Table 4: The differences in the detection rates of different multimedia types of packets before and after optimization

multimedia types	P	number of packets	the detection rate before optimization	the detection rate after optimization	Changes
octet-stream	2.4	31	22%	100%	78%↑
x-javascript	2.2	35	24%	86%	62%↑
html	1.5	43	26%	20%	-6%↓
jpeg	1.3	58	34%	18%	-16%↓
gif	1.3	156	71%	32%	-39%↓
x-shockwave-flash	1.7	27	18%	29%	11%↑
.....	...	...	...	...	...

Table 5: The comparison of the number of detected multimedia packets with different media types

Multimedia types carried by packets	before optimization	after optimization	The increased proportion
octet-stream	28	31	9.6%
x-javascript	65	72	9.3%
html	66	70	5.7%
jpeg	110	125	10.7%
gif	351	403	12.6%
x-shockwave-flash	7	7	0%
...	...	...	...

ble 4).

As can be seen from Table 4, the detection rate of multimedia packets with higher degree of risk (P shown in Table 1) has been generally improved after optimization. For example, the octet-stream type has increased by 78% and the x-JavaScript type has increased by 62%. In addition to the effect of the optimization, the reason for its great improvement is that the detection rate is low before optimization because these multimedia packets have been selected randomly to be detected. On the other hand, it shows that the detection rate of multimedia packets with lower degree of risk has decreased. For example, the detection rate of "gif" packets decreases by 39%.

As Table 5 shows, since the use of the Multimedia Packets Processing Method with 0/1 Knapsack Problem in NIDS, the amount of processed multimedia packets with risky information between W1 and W3 has increased obviously, among which such types as gif and jpeg appearing more frequently in the flow have greater increased, which increase respectively by 10.7% and by 12.6%. However, those multimedia types appearing less frequently such as x-shockwave-flash type has not increased.

The third step in the experiment is to compare the change of the packet loss rate before and after optimization (as shown in Figure 4).

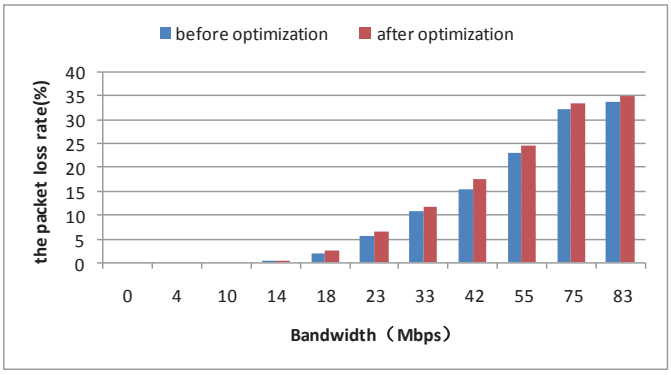


Figure 4: The contrast of packet loss rate before and after optimization

As is shown in Figure 4, the change of the packet loss rate before and after optimization is not very obvious. The reason for its slightly higher packet loss rate after optimization is its consumption of system resource in the

optimal choice sequence of multimedia packets.

6 Conclusion

Among many studies on NIDS, this paper started from the study of multimedia file in the network. On the basis of proposed Multimedia Packets Processing Method, this paper mainly focuses on adding the idea of optimization in 0/1 Knapsack Problem to the main decision-making steps of Multimedia Packets Processing Method. By means of above improvement, within a network flow range, NIDS can make limited processing capability focus on the more risky multimedia packets. Various experiments have shown that the method can effectively improve the detective rate of multimedia packets with dangerous information.

Acknowledgment

This work was supported by Science and Technology Planning Project of Shannxi Province of China (No.2014K05-43).

References

[1] M. Arun, A. Krishnan, "Functional verification of signature detection architectures for high speed network applications," *International Journal of Automation and Computing*, vol. 9, no. 4, pp. 301, 2012.

[2] S. Dong, X. Li, and Z. Yin, "Improved string matching algorithm", *Computer Engineering and Applications*, vol. 49, no. 8, pp. 133-137, 2013.

[3] S. Jians, D. Wang, "An improved ant colony clustering method for intrusion detection," *Computer Technology and Development*, vol. 23, no. 12, pp. 139-142, 2013.

[4] Z. Li, Y. Li, and L. Xu, "Research of intrusion detection method based on particle swarm optimization and immune Agent," *Computer Engineering and Applications*, vol. 48, no. 1, pp. 102-104, 2012.

[5] F. Z. Liu, "A clustering method for anomaly intrusion detection," *Computer Security*, vol. 15, no. 2, pp. 156-161, 2013.

[6] W. G. Liu, Y. G. Hu, "DHSWM: An improved multi-pattern matching algorithm based on WM algorithm," *Journal of Central South University*, vol. 4212, pp. 3765-3771, 2011.

[7] L. Lu, T. Ye, "Research on proving multi-pattern matching algorithm based on deterministic finite-state automation," *Computer Applications and Software*, vol. 30, no. 7, pp. 321-323, 2013.

[8] M. Mahoney, P. K. Chan, "An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection," RAID , pp. 220-237, 2003.

[9] S. Pan, T. Morris, and U. Adhikari, "A specification-based intrusion detection framework for cyber-physical environment in electric power system," *International Journal of Network Security*, vol.17, no.2, pp.174-188, Mar. 2015.

[10] Q. Qian, T. Wang, and R. Zhang, "Relative network entropy based clustering algorithm for intrusion detection," *International Journal of Network Security*, vol.15, no.1, pp.16-22, Jan. 2013.

[11] Q. Quan, C. J. Xiao, and R. Zhang, "Grid-based data stream clustering for intrusion detection," *International Journal of Network Security*, vol.15, no.1, pp.1-8, Jan. 2013.

[12] Z. Shi, Y. Xia, F. Wu, and J. Dai, "The discretization algorithm for rough data and its application to intrusion detection," *Journal of networks*, vol. 9, no. 6, pp. 1380-1387, 2014.

[13] T. Song, D. N. Li, "Memory efficient algorithm and architecture for multi-pattern matching," *Journal of Software*, vol. 24, no. 7, pp. 1650-1665, 2013.

[14] K. Sravani, P. Srinivasu, "Comparative study of machine learning algorithm for intrusion detection system," *Advances in Intelligent Systems and Computing*, vol. 247, pp.189-196, 2014.

[15] M. Uddin, A. A. Rehman, N. Uddin, J. Memon, R. Alsaqour, and S. Kazi," Signature-based multi-layer distributed intrusion detection system using mobile agents," *International Journal of Network Security*, vol.15, no.2, pp.97-105, Mar. 2013.

[16] Q. Wu, J. Cao, and R. Zheng, et al. "Intrusion feature selection algorithm based on particle swarm optimization," *Computer Engineering and Applications*, vol. 49, no. 7, pp. 89-92, 2013.

[17] H. Yan, "Research on improved BMH single-pattern matching algorithm based on Snort," *Computer Engineering and Applications*, vol. 48, no. 31, pp. 78-81, 2012.

[18] G. Zhai, "Intrusion detection algorithm based on fuzzy evaluation and clustering analysis," *Computer Engineering and Applications*, vol. 48, no. 21, pp. 99-102, 2012.

[19] X. Y. Zhang, "Research of intrusion detection system dataset-KDD CUP99," *Computer Engineering and Design*, vol. 31, no. 22 , pp. 4809-4812, 2010.

[20] X. Zhao, "Research on Dynamic self-adapting multimedia data processing method based on Snort". *Computer Systems & Applications*, Vol. 20, No. 4, pp. 211-213,2011

[21] X. Zhao, C. Wang, "The improvements to snort intrusion detection system," *Journal of Xi'an Polytechnic University*, vol. 21, no. 6, pp. 859-863, 2006.

Xu Zhao is a Lecturer in the Department of Computer Science, Xi'an Polytechnic University, Shannxi, China. He received the M.E. degree from Xian Electronic Technology University, Xian City, Shannxi Province, China in 2007. He has some projects in research supported by provincial funds. His research interest is Network Security.

An Authenticated Privacy-preserving Attribute Matchmaking Protocol for Mobile Social Networks

Solomon Sarpong[1], Chunxiang Xu[2], and Xiaojun Zhang[3]
(Corresponding author: Solomon Sarpong)
Department of Computer Science, University of Electronic Science and Technology of China, Chengdu, China
Main Building A1-406, No. 2006, Xiyuan Avenue, West Hi-Tech Zone, Chengdu 611731, China
(Email: sarpong.uestc@gmail.com)
(Received Nov. 5, 2014; revised and accepted Jan. 16 & Mar. 4, 2015)

Abstract

Matchmaking on mobile social networks has evolved over the years since its advent. On MSN, important, personal, private and sensitive information are shared. This has necessitated the need for efficient and privacy-preserving matchmaking protocols so as to prevent unintended persons from accessing such information. In most of the existing matchmaking protocols the inputs are private and personal. Hence malicious participants may choose their sets arbitrarily so as to learn more information about the input of an honest individual. As an improvement on the existing matchmaking protocols, we propose an authenticated hybrid matchmaking protocol that will help match-pair seekers find the most appropriate pair without leaking private and sensitive information to unintended persons. In our protocol, attributes used are certified by a certification authority. Also, the initiator sets a threshold number of common attributes that an individual should have to qualify as a match-pair. With the use of certification of attributes to ensure that the inputs are not arbitrary and a preset threshold number of common attributes defined, an initiator can adequately find the best pair(s) without leaking any private and sensitive information. Furthermore, our protocol has the ability to resist semi-honest and malicious attacks.

Keywords: Authentication, certification, matchmaking, nonspoofability, user-profiling

1 Introduction

Social networks are highly dynamic applications; their modifications come very quickly over time through the addition of new edges, signifying the appearance of new interactions in the underlying social structure. Understanding the mechanisms by which they evolve is a fundamental question that is still not well understood [16]. Data sharing on social networks is finding ever-growing broad applications and becoming an essential part of our daily life. It enables real time communications such as instant messaging. Despite the various appealing features offered, users′ data privacy are always at risk when the network is exploited for adversarial activities, e.g., accessing private data without permissions, illegally selling private data, profiling the data owner, etc. The risk is dramatically increased especially when users are encouraged to include their real names. This makes users vulnerable to data privacy breaches. However, Chen et al. [5] observed that individuals can protect their own private or sensitive information by restricting the intended purpose of data access by denying the right to access for some purposes.

A private matchmaking protocol allows two or more mutually suspicious parties with matching credentials to locate and authenticate each other without revealing their credentials or identities to anyone including a matchmaker. Private matchmaking is more than mutual authentication of suspicious parties in that it has further requirements on privacy [23]. In private matchmaking protocols, most often than not the sharing of information involves two parties seeking to know if their private sets have any information in common. Hence, the two main challenges encountered are; (1) how to enable this type of sharing such that the parties learn no (or minimal) information beyond what they are entitled to and (2) how to do so efficiently in real world practical terms [11]. Furthermore, in the scenario where two or more companies want to identify their common customers, they would like to ensure that (1) neither party learns more than their own data and must obtain the intersection (if one exists), while neither should learn anything about the others′ set and (2) they should learn the results of the intersection mutually. These are the premise of private set intersection.

Private Set Intersection (PSI) is a cryptographic pro-

tocol that involves two players, say Alice and Bob, each with a private set. Their goal is to compute the intersection of their respective sets, such that minimal information is revealed in the process. In other words, Alice and Bob should learn the elements in the intersection (if any) and nothing else. Ideally, this should be a mutual process thereby neither party has any advantage over the other Li et al. [14], Xie and Hengartner [28] and FindU [18]. But in [1, 10, 15], the protocols are asymmetric. Asymmetric private set intersection protocols may be acceptable or even desirable in some scenarios, but may be undesirable in others. In the likely event of two dishonest parties in a protocol, the one receiving the intersection may not truly report the intersection leading to information asymmetry.

Alice and Bob may hold sets S_A and S_B, respectively and may want to compute the intersection of their set. Their wish is to jointly compute the intersection in such a way that reveals as little as possible about S_A to Bob and S_B to Alice. In other words, both Alice and Bob should learn only $S_A \cap S_B$ but nothing more. While this task could be completed with general secure multiparty techniques, it is far more efficient to have a dedicated protocol. Also, in both asymmetric and symmetric protocols, since each party is not willing to disclose the content of their list, ordinary private set intersection will not be appropriate to use in finding the intersection. In light of these, authorized private set intersection is more appropriate.

Authorized Private Set Intersection (APSI) and its variants [3, 4, 8] ensure that each party can only use elements certified by a trusted authority in the intersection protocol. In particular, we consider the scenario where two parties each hold a set of elements and wish to find the intersection of their elements without revealing other elements that are not in the intersection. In such applications, it is important to ensure that each data item being exchanged is properly authenticated or authorized by a trusted authority in the intersection protocol [8]. When authorization is done, it thwarts dishonest behavior. Unless some form of authentication is required, a malicious party can claim possession of fictitious data items, in an attempt to find out whether the other party possesses those data items. The problem of authentication of mutually suspicious parties is becoming more and more important with the proliferation of distributed systems. A user in a distributed system may not only need to verify the identity of the system, but may require that the system, or another user or node in the system, verifies itself to him/her. Moreover, both sides may require some degree of authentication before they release any information about themselves [7]. The goal of authorizing the private sets of users is to restrict their inputs. This reduces the strength of a malicious attacks by users.

As a contribution to research, we are proposing an authenticated hybrid matchmaking protocol that is very efficient, privacy preserving and secure against malicious attacks. In the proposed protocol, not only does the initiator find a matching-pair, but the pair that meets a preset minimum number of common attributes set by the initiator. Furthermore, apart from the matched-pair that is privy to the type of common attributes, no one else does.

The rest of this paper is organized as follows: in Section 2, we present related work. Our protocol and algorithm for the matchmaking is presented in Section 3. In Section 4, we present the security and experimental implementation. Finally, we conclude this paper in Section 5.

2 Related Work

In 1985, Baldwin and Gramlich [2] laid the foundation for matchmaking in social network. In their paper, there was the usage of a trusted third party in the matchmaking protocol. Meldew [21] proposed a more efficient matchmaking protocol that needed no trusted third party to be continuously available. Zhang and Needham [31] also proposed a matchmaking protocol that depends on the availability of a public database service to all users. However, the security of this protocol depends on the security of the hash function and the encryption algorithm used. Freedman, Nissim and Pinkas [10] also considered the problem of computing the intersection of private data sets of two parties, where the data sets contain lists of elements taken from a large domain.

Shin and Gligor [25] sort to achieve anonymity of protocol users and authentication of wish matches with new security goals, which appear to be fundamental to private matchmaking. Hence, the overall set of security goals of this protocol comprises: authenticity of users and wish matches; privacy resistance to off-line dictionary attacks and forward privacy of users′ identities and their wishes. Sang and Shen [22] observed that when data sets are distributed on different sources, finding out their intersection while preserving the privacy of the data sets is a widely required task in set intersection. In their paper, they addressed privacy preserving set intersection (PPSI) problem, in which each of the n parties learns no elements other than the intersection of their n private data sets.

Using identity based encryption (IBE), Shamir [24] formulated a novel scheme which enables the user to sign and encrypt the messages s/he sends and to decrypt and verify the messages s/he receives in a totally independent way, regardless of the identity of the other party. The scheme also enables any pair of users to communicate securely and to verify each other's signatures without exchanging private or public keys, without keeping key directories and without using the services of a third party. Camenisch et al. [3], proposed the searchable encryption scheme that provides an important mechanism to cryptographically protect data while keeping it available to be searched and accessed for matching information. In the scheme, they proposed two encryptions; public key encryptions with oblivious keyword search (PEOKS) and committed blind anonymous identity-based encryption (IBE). Lin et al. [17] proposed efficient blind-key encryption pro-

tocols for anonymous identity-based encryption and an anonymous hierarchical identity-based encryption. These schemes were used in privacy preserving profiles searching (PPPS) problem.

Sun et al. [26] proposed a privacy-preserving scheme for data sharing in social networks with efficient revocation for deterring a contact′s access right to the private data once the contact is removed from the social group. Zhang et al. [30] also propose a privacy-preserving verifiable profile matching scheme which is based on symmetric cryptosystem and thus improves efficiency. It relies on a pre-determined ordered set of attributes and uses it as a common secret shared by users. However, the scheme is not applicable to unordered sets of attributes such as random capabilities. Cristofaro and Tsudik [8] considered several flavors of Private Set Intersection (PSI) and constructed some provably secure protocols. They proposed efficient protocols for plain and authorized private set intersection and noted that, the choice between them depends on whether there is a need for client authorization and server unlinkability, as well as on server's ability to engage in pre-computation.

Matchmaking is an integral aspect of private set intersection. Matchmaking protocol is a private set intersection problem in which match-pair is made by computing the intersection of their individual attributes. One of the techniques in matchmaking protocols is the use of trusted third party. This technique can be found in protocol applications such as [9, 12, 13]. However, the use of trusted third party in matchmaking protocols has well-known problems.

Another is the fully distributed technique, which requires no trusted server in the whole matchmaking process [18, 19, 29]. The operations, such as the distribution of personal attributes data, the computation of the intersection set, and the dissemination of results are performed among multi-parties, without any trusted third party. The attributes of the initiator and the candidates are shared among multi-parties using Shamir Secret Sharing Scheme, the computation of common attributes set are conducted among multi-parties as well.

The third technique in use is a hybrid, where a trusted centralized server is needed only for the purpose of management and verification, and it does not participate in the matchmaking operations. This mechanism can provide efficient matchmaking services with relatively high scalability. In [6, 20, 27, 28], are protocols based on hybrid mechanisms designed to support privacy preserving attributes matchmaking functions for mobile social networks.

In [1], the protocol allows only one party, Alice to compute the intersection. Alice may decide; (1) not to report the intersection set truthfully to the other user, Bob; (2) to discontinue with the matchmaking after knowing the attributes of Bob. Furthermore as the proposed protocol is one way, several malicious attacks can be launched the protocol users. Xie and Hengartner [28] improved on this protocol by removing the likelihood of malicious attack by persons involved in the protocol. Also, both persons in the protocol perform the intersection set.

To the best of our knowledge, most of the existing proposed matchmaking protocols do not take into consideration if the would-be pair has enough common attributes to qualify to be paired. However, in matchmaking protocols as found in [14, 18, 27] there was an improvement in the matchmaking protocols by letting a user (called initiator) find the best match among multi-parties (called candidates). In these protocols, the best match is the user (among other candidates) who has the maximum intersection set size with the initiator. It can be observed that the best match does not necessarily mean the pair has got enough common attributes to make a good pair. However, in [17], the initiator sets a threshold number of common attributes that a user should have to qualify as a pair. The number of attributes that users in the protocol have in common is assessed by the use of privacy-preserving scalar product computation. When the number of attributes they have in common is at least the preset threshold, they become a match-pair.

3 Matchmaking Protocol

A certification authority, CA that cannot be compromised, an initiator looking for matching-pair(s) and a number of persons(candidates) that the initiator is looking for matching-pair(s) from constitute our protocol. This protocol will help match-pair seekers find the most appropriate pair in a mutual matchmaking protocol. The protocol users can only know the size of the intersection (if it exits), his/her input to the matching and the actual attributes they have in common if a pair is found. Apart from these information, nothing else is available to a user(s). For this protocol to maintain its security and privacy, users should keep their private keys safe, so that malicious persons cannot steal their private keys to impersonate them. Also, there should be trust between the matched-pair so that attributes of the pair will not be disclosed to others.

3.1 Initial Phase

Assuming there are T users, each possessing a portable device; $A_{Threshold}$ is the threshold number of attributes set by the initiator; communication among users is done through Bluetooth or Wifi on users′ portable devices. For the initiator to find a match-pair, the number of common attributes should be at least $A_{Threshold}$. The initiator, Alice possesses a set of attributes $a = \{a_1, a_2, \ldots, a_k\}$ and each candidate also possesses the set of attributes $b_t = \{b_{t1}, b_{t2}, \ldots, b_{tp}\}$, $t = 1, \ldots, m$. In the matchmaking, two attributes are exactly the same if they are the same semantically.

Table 1: Notations

Notation	Explanation
R_A	Random number chosen by Alice, $R_A \longleftarrow_r Z_{\lfloor\sqrt{N}/4\rfloor}$
R_t	Random number chosen by each candidate, $R_t \longleftarrow_r Z_{\lfloor\sqrt{N}/4\rfloor}$
ID_t	Identity of each candidate
ID_A	Identity of Alice
a^e	Exponentiated attributes of Alice
b_t^e	Exponentiated attributes of each candidate
S_l	Computation to certify Alice′s attributes, $S_l = H(ID_A \parallel a_l)^d \mathrm{mod} N$
σ_{th}	Computation to certify each candidate′s attributes, $\sigma_{th} = H(ID_t \parallel b_{th})^d \mathrm{mod} N$
ζ_A	Random permutation of Alice
ζ_t	Random permutation of each candidate
$\mid I_{At} \mid$	Number of attributes that Alice and each candidate have in common
$\mid I_{tA} \mid$	Number of attributes that each candidate and Alice have in common

3.2 Keys Generation

RSA key pair, (e, d) and $N = pq$, where p and q are large prime numbers generated by the CA. The CA makes N, e and a collision resistant cryptographic hash function H, public. RSA key pairs (e_T, d_T) are created by each user, who then makes e_T public. Username and an ID, which is the hash of his/her RSA private key are also created by each user.

3.3 Attributes Certification

The attributes of Alice and the t candidates are $a = \{a_1, a_2, \ldots, a_k\}$ and $b_t = \{b_{t1}, b_{t2}, \ldots, b_{tp}\}$, $t = 1, \ldots, m$ respectively. Alice′s attributes becomes $a^e = \{a_1^e, a_2^e, \ldots, a_k^e\}$ after, she exponentiates her attributes using the public key of the CA. Also, the attributes of each candidate becomes $b_t^e = \{b_{t1}^e, b_{t2}^e, \ldots, b_{tp}^e\}$ after, exponentiating the attributes with the public key of the CA. Each user then encrypts his/her attributes, ID, username, and the public key pair of his/her RSA key using the public key of the CA and sends it to the CA. Alice sends $E_e\{a^e \parallel ID_A \parallel username \parallel RSApublickey, e_A\}$ to the CA. Each candidate also sends $E_e\{b_t^e \parallel ID_t \parallel username \parallel RSApublickey, e_t\}$ to the CA. The CA certifies the attributes and returns $A = \{(a_1, S_1), (a_2, S_2), \ldots, (a_k, S_k)\}$, where $S_l = H(ID_A \parallel a_l)^d \mathrm{mod} N$ and $B_t = \{(b_{t1}, \sigma_{t1}), (b_{t2}, \sigma_{t2}), \ldots, (b_{tp}, \sigma_{tp})\}$ where $\sigma_{th} = H(ID_t \parallel b_{th})^d \mathrm{mod} N$ to Alice and each candidate respectively.

3.4 Matchmaking Phase

The private input of Alice and each candidate after the certification becomes $A = \{(a_1, S_1), (a_2, S_2), \cdots, (a_k, S_k)\}$ and $B_t = \{(b_{t1}, \sigma_{t1}), (b_{t2}, \sigma_{t2}), \cdots, (b_{th}, \sigma_{th})\}$ respectively. Alice chooses a random number $R_{A:l} \longleftarrow_r Z_{\lfloor\sqrt{N}/4\rfloor}$ and computes $M_{A:l} = S_l \cdot g^{R_{A:l}} \mathrm{mod} N$. Alice then sends $MES_1 = M_{A:l}; l = 1, \ldots, n$ to each candidate. Each candidate also chooses a random number $R_{B_t:h} \longleftarrow_r Z_{\lfloor\sqrt{N}/4\rfloor}$ and computes $M_{B_{t:h}} = \sigma_{th} \cdot g^{R_{B_{t:h}}} \mathrm{mod} N$. Each candidate then sends $MES_2 = M_{B_t:h}; t = 1, \ldots, m; h = 1, \ldots, p$ to Alice.

Alice further chooses and stores another random number $R_A \longleftarrow_r Z_{\lfloor\sqrt{N}/4\rfloor}$. Alice computes $Z_A = g^{eR_A} \mathrm{mod} N$, $M'_{B_t:h} = (M_{B_t:h})^{eR_A} \mathrm{mod} N$ and $\{a_1, a_2, \ldots, a_k\}^{R_A}$. Alice chooses a random permutation ζ_A and computes $K_{A:l} = \zeta_A\{a_1, a_2, \ldots, a_k\}^{R_A}$. Alice then sends $MES_3 = Z_A \parallel M_{B_t:h}, t = 1, \ldots, m; h = 1, \ldots, p \parallel K_{A:l}, l = 1, \ldots, k$ to each candidate. Also, each candidate chooses and stores another random number $R_t \longleftarrow_r Z_{\lfloor\sqrt{N}/4\rfloor}$. Each candidate computes $Z_t = g^{eR_t} \mathrm{mod} N$, $M'_{A:l} = (M_{A:l})^{eR_t} \mathrm{mod} N$ and $\{b_{t1}^{eR_t}, b_{t2}^{eR_t}, \ldots, b_{tp}^{eR_t}\}$. Each candidate chooses a random permutation ζ_t and computes $K_{B_t:h} = \zeta_t\{b_{t1}^{R_t}, b_{t2}^{R_t}, \ldots, b_{tp}^{R_t}\}$. Each candidate then sends $MES_4 = Z_B \parallel M_{A_i}, l = 1, \ldots, k \parallel K_{B_t:h}, t = 1, \ldots, m; h = 1, \ldots, p$ to Alice.

In Step 7, Alice signs her ID, together with MES_1, MES_2, MES_3 and MES_4 and sends to each candidate. Alice sends $Sig_{d_A}(ID_{Alice} \parallel MES_1 \parallel MES_2 \parallel MES_3 \parallel MES_4)$ to each candidate. Each candidate also signs the ID, together with MES_1, MES_2, MES_3 and MES_4 and sends to Alice. Thus, each candidate sends $Sig_{d_t}(ID_t \parallel MES_1 \parallel MES_2 \parallel MES_3 \parallel MES_4)$ to Alice. Alice and each candidate then checks if the MES_1, MES_2, MES_3 and MES_4 received from each other is the same as those computed or received earlier on in the protocol.

Alice sends her random number to each candidate by sending $Sig_{d_A}(ID_{Alice} \parallel ID_t \parallel R_A)$. Likewise, each candidate also sends his/her random number to Alice by sending $Sig_{d_t}(ID_t \parallel ID_{Alice} \parallel R_t)$. Alice then computes and form a list $K_{A:l} = \zeta_A\{a_1^{R_A R_t}, a_2^{R_A R_t}, \ldots, a_k^{R_A R_t}\}$ which she sends to each candidate. Each candidate also computes and forms a list $K_{B_t:h} = \zeta_t\{b_{t1}^{R_t R_A}, b_{t2}^{R_t R_A}, \ldots, b_{tp}^{R_t R_A}\}$ and sends to Alice. Alice then computes and outputs the number of common attributes $|I_{At}|$ in $A \cap B_t$ s. t. $|I_{At}| \in K_{A:l} \cap K_{B_t:h}$, $\forall\ l = 1, \ldots, k; t = 1, \ldots, m; h = 1, \ldots, p$. Each candi-

date also computes and outputs the number of common attributes $|I_{tA}|$ in $A \cap B_t$ s. t. $|I_{tA}| \in K_{A:l} \cap K_{B_t:h}$, $\forall\ l = 1, \ldots, k; t = 1, \ldots, m; h = 1, \ldots, p$.

Among the t candidates, let Bob be the only candidate with $|I_{tA}| \geq A_{Threshold}$. Hence, Bob becomes a match-pair of Alice. Alice and Bob at this point know only the number of attributes they have in common. In order for them to know the actual attributes they have in common, they exchange their random permutations. By sending $E_{e_B}(\zeta_A)$ to Bob, Alice thus sends her random permutation to Bob. Also, Bob sends his random permutation to Alice by sending $E_{e_A}(\zeta_B)$. Alice knowing ζ_B, computes ζ_B^{-1} and recovers $\{b_1^{R_B R_A}, b_2^{R_B R_A}, \ldots, b_p^{R_B R_A}\}$ from $\zeta_B\{b_1^{R_B R_A}, b_2^{R_B R_A}, \ldots, b_p^{R_B R_A}\}$. Also, Bob knowing ζ_A computes ζ_A^{-1} and recovers $\{a_1^{R_A R_B}, a_2^{R_A R_B}, \ldots, a_k^{R_A R_B}\}$ from $\zeta_A\{a_1^{R_A R_B}, a_2^{R_A R_B}, \ldots, a_k^{R_A R_B}\}$. Alice and Bob will then know the actual attributes they have in common.

3.5 The Algorithm

The algorithm enables the initiator, Alice to find a candidate(s) who has the minimum threshold number of common attributes.

4 Security Analysis

The CA certifies all the attributes that are used. As depicted in the algorithm, for each of the attributes that Alice sends to the CA, the CA computes $S_l = H(ID_A \parallel a_l)^d \mathrm{mod} N$. Likewise, the CA certifies the attributes that each candidate uses by computing $\sigma_{th} = H(ID_t \parallel b_{th})^d \mathrm{mod} N$. The certification of attributes ensures that the attributes of the persons in the protocol are bound to them. They therefore cannot change or modify their attributes so as to gain more information from the others. This to a large extent, will eliminate semi-honest and malicious attacks by persons in the protocol.

In Step 5 of the algorithm, Alice computes $\zeta_A\{a_1, a_2, \ldots, a_k\}^{R_A}$. The computation of $\zeta_A\{a_1, a_2, \ldots, a_k\}^{R_A}$ makes it computationally impossible for any candidate to map $a_l^{R_A}$, $l = 1, \ldots k$ to the corresponding attribute in $\zeta_A\{a_1, a_2, \ldots, a_k\}^{R_A}$ in polynomial time. Hence in Step 9, there is no way a candidate can know the actual attributes of Alice. Likewise, the computation of $\zeta_t\{b_{t1}, b_{t2}, \ldots, b_{tp}\}^{R_t}, t = 1, \ldots, m; h = 1, \ldots, p$ in Step 6, makes it computationally impossible for Alice to map $b_{th}^{R_t}$ to the corresponding attribute in $\zeta_t\{b_{t1}, b_{t2}, \ldots, b_{tp}\}^{R_t}$ in polynomial time. Hence in Step 9, there is no way Alice can know the actual attributes of any of the candidate(s).

Also, the computation of $K_{A:l} = \zeta_A\{a_1^{R_A R_t}, a_2^{R_A R_t}, \ldots, a_k^{R_A R_t}\}$, $\forall\ l = 1, \ldots, k$ and $K_{B_t:h} = \zeta_t\{b_{t1}^{R_t R_A}, b_{t2}^{R_t R_A}, \ldots, b_{tp}^{R_t R_A}\}$ $\forall\ t = 1, \ldots, m$ and $h = 1, \ldots, p$ by Alice and each candidate is to ensure that even if an adversary happens to know a user′s random number, the personal attributes will not be known. Furthermore, in Step 7, Alice sends $Sign_{d_A}(ID_{Alice} \parallel MES_1 \parallel MES_2 \parallel MES_3 \parallel MES_4)$ to each candidate; likewise, each candidate also sends $Sign_{d_t}(ID_t \parallel MES_1 \parallel MES_2 \parallel MES_3 \parallel MES_4)$ to Alice. In this step, each user in the protocol authenticates and confirms the correctness of the values received. Thus, each user checks if the values received in Step 7 from each other is the same as the values computed or received earlier on in the algorithm. In the event that any user observes that any of the values computed or received earlier in the algorithm is not the same as the values received in Step 7, the algorithm is terminated.

Algorithm 1 Computing the Number of Common Attributes

Require: Let $\{N, e, g, H\}$ be inputs common to Alice and the candidates from the CA.

1: Private attributes of Alice, $(a_1, a_2, \ldots, a_k)$; After certification, Alice′s private set becomes $A = \{(a_1, S_1), (a_2, S_2), \ldots, (a_k, S_k)\}$, where $S_l = H(ID_A \parallel a_l)^d \mathrm{mod} N$.

2: The t candidates with h attributes have private input, $(b_{t1}, b_{t2}, \ldots, b_{tp})$; After certification, their private input set becomes $B_t = \{(b_{t1}, \sigma_{t1}), (b_{t2}, \sigma_{t2}), \ldots, (b_{tp}, \sigma_{tp})\}$, where $\sigma_{th} = H(ID_t \parallel b_{th})^d \mathrm{mod} N$.

3: For all $l = 1, \ldots, k$, Alice chooses a random number $R_{A:l} \longleftarrow_r Z_{\lfloor \sqrt{N}/4 \rfloor}$ and computes $M_{A:l} = S_l \cdot g^{R_{A:l}} \mathrm{mod} N$. Alice then sends $MES_1 = M_{A:l}, l = 1, \ldots, k$ to each candidate.

4: For all $t = 1, \ldots, m$ and $h = 1, \ldots, p$, each candidate chooses a random number $R_{B_t:h} \longleftarrow_r Z_{\lfloor \sqrt{N}/4 \rfloor}$ and computes $M_{B_t:h} = \sigma_{th} \cdot g^{R_{B_t:h}} \mathrm{mod} N$. Each candidate then sends $MES_2 = M_{B_t:h}, t = 1, \ldots, m$, $h = 1, \ldots, p$ to Alice.

5: Alice further chooses and stores another random number, $R_A \longleftarrow_r Z_{\lfloor \sqrt{N}/4 \rfloor}$, computes $Z_{A:l} = g^{eR_A} \mathrm{mod} N$. Alice further computes $M'_{B_t:h} = (M_{B_t:h})^{eR_A} \mathrm{mod} N$ for all $t = 1, \ldots, m$ and $h = 1, \ldots, p$. Alice computes $\{a_1, a_2, \ldots, a_k\}^{R_A}, l = 1, \ldots, k$ and randomly permutes it using a random permutation ζ_A. Alice then sends $MES_3 = Z_A \parallel M'_{B_t:h}, t = 1, \ldots, m; h = 1, \ldots, p \parallel \zeta_A\{a_1, a_2, \ldots, a_k\}^{R_A}, l = 1, \ldots, k$ to each candidate.

6: Also, each candidate further chooses and stores another random number, $R_t \longleftarrow_r Z_{\lfloor \sqrt{N}/4 \rfloor}$ computes $Z_{B_t:h} = g^{eR_t} \mathrm{mod} N$ and $M'_{A:l} = (M_{A:l})^{eR_t} \mathrm{mod} N$ for all $l = 1, \ldots, k$ and $t = 1, \ldots, m$. Each individual computes $\{b_{t1}, b_{t2}, \ldots, b_{tp}\}^{R_t}$ and randomly permutes it using the random permutation ζ_t. Each candidate then sends $MES_4 = Z_t \parallel M'_{A:l}, l = 1, \ldots, k \parallel \zeta_t\{b_{t1}, b_{t2}, \ldots, b_{tp}\}^{R_t}$, $t = 1, \ldots, m$ and $h = 1, \ldots, p$, to Alice.

7: Alice using her private key signs and sends $Sign_{d_A}(ID_{Alice} \parallel MES_1 \parallel MES_2 \parallel MES_3 \parallel MES_4)$ to each candidate. Each candidate also using his/her private key signs and sends $Sign_{d_t}(ID_t \parallel MES_1 \parallel MES_2 \parallel MES_3 \parallel MES_4)$ to Alice.

8: Alice as well as each candidate then verifies if MES_1, MES_2, MES_3 and MES_4 received in Step 7 is the same as those computed or received in the previous steps of the algorithm.

9: If Step 8 is verified correctly, Alice then sends $Sign_{d_A}(ID_{Alice} \parallel ID_t \parallel R_A)$ to each candidate. Each candidate also sends $Sign_{d_t}(ID_t \parallel ID_{Alice} \parallel R_t)$ to Alice.

10: Alice then computes and forms a list, $K_{A:l} = \zeta_A\{a_1^{R_A R_t}, a_2^{R_A R_t}, \ldots, a_k^{R_A R_t}\}$, $l = 1, \ldots, k$ and each of the candidates also computes and forms a list, $K_{B_t:h} = \zeta_t\{b_{t1}^{R_t R_A}, b_{t2}^{R_t R_A}, \ldots, b_{tp}^{R_t R_A}\}$, $t = 1, \ldots, m$ and $h = 1, \ldots, p$.

11: Alice then sends $K_{A:l} = \zeta_A\{a_1^{R_A R_t}, a_2^{R_A R_t}, \ldots, a_k^{R_A R_t}\}$, $l = 1, \ldots, k$ to each candidate. Also, each candidate sends $K_{B_t:k} = \zeta_t\{b_{t1}^{R_t R_A}, b_{t2}^{R_t R_A}, \ldots, b_{tp}^{R_t R_A}\}$, $t = 1, \ldots, m$ and $h = 1, \ldots, p$ to Alice.

12: Alice computes and outputs the number of common attributes $|\ I_{At}\ |$ in $A \cap B_t$ such that $|\ I_{At}\ | \in K_{A:l} \cap K_{B_t:h}$, $\forall\ l = 1, \ldots, k; t = 1, \ldots, m$ and $h = 1, \ldots, p$. Hence, the number of attributes Alice has in common with each candidate is $|\ I_{At}\ | = \zeta_A\{a_1^{R_A R_t}, a_2^{R_A R_t}, \ldots, a_n^{R_A R_t}\} \cap \zeta_t\{b_{t1}^{R_t R_A}, b_{t2}^{R_t R_A}, \ldots, b_{tp}^{R_t R_A}\}$.

13: Each candidate also computes and outputs the number of common attributes $|\ I_{tA}\ |$ in $A \cap B_t$ such that $|\ I_{tA}\ | \in K_{B_t:h} \cap K_{A:l}$, $\forall\ i = 1, \ldots, k; t = 1, \ldots, m$ and $h = 1, \ldots, p$. The number of attributes each candidate has in common with Alice is $|\ I_{At}\ | = \zeta_A\{a_1^{R_A R_t}, a_2^{R_A R_t}, \ldots, a_n^{R_A R_t}\} \cap \zeta_t\{b_{t1}^{R_t R_A}, b_{t2}^{R_t R_A}, \ldots, b_{tp}^{R_t R_A}\}$.

This is then reported to the CA, who then checks to find out the person who has cheated. Hence, this step checks malicious attacks by protocol users.

4.1 Correctness of the Algorithm

In Step 11 of the algorithm, Alice computes and sends $K_{A:l} = \zeta_A\{a_1^{R_AR_t}, a_2^{R_AR_t}, \ldots, a_k^{R_AR_t}\}$, $l = 1, \ldots, k$ to each candidate. Each candidate also computes and sends $K_{B_t:k} = \zeta_t\{b_{j1}^{R_tR_A}, b_{t2}^{R_tR_A}, \ldots, b_{tp}^{R_tR_A}\}$, $t = 1, \ldots, m$ and $h = 1, \ldots, p$ to Alice. At the end of the algorithm, Alice outputs;
$|I_{At}| \in \zeta_A\{a_1^{R_AR_t}, a_2^{R_AR_t}, \ldots, a_k^{R_AR_j}\} \cap \zeta_t\{b_{t1}^{R_tR_A}, b_{t2}^{R_tR_A}, \ldots, b_{tp}^{R_tR_A}\}$.
Each candidate also outputs;
$|I_{tA}| \in \zeta_A\{a_1^{R_AR_t}, a_2^{R_AR_t}, \ldots, a_k^{R_AR_j}\} \cap \zeta_t\{b_{t1}^{R_tR_A}, b_{t2}^{R_tR_A}, \ldots, b_{tp}^{R_tR_A}\}$.

Since $|I_{At}| = |I_{tA}|$, the number of attributes in both $|I_{At}|$ and $|I_{tA}|$ are the same. Hence, the algorithm is correct.

4.2 Achievement of Privacy in the Algorithm

The algorithm achieves privacy for the users′. It can be observed that at the end of the algorithm, Alice outputs $| I_{At} | \in \zeta_A\{a_1^{R_AR_t}, a_2^{R_AR_t}, \ldots, a_k^{R_AR_t}\} \cap \zeta_t\{b_{t1}^{R_tR_A}, b_{t2}^{R_tR_A}, \ldots, b_{tp}^{R_tR_A}\}$, $\forall\ l = 1, \ldots, k; t = 1, \ldots, m$ and $h = 1, \ldots, p$. This only allows her to know the number of attributes she has in common with each candidate. In like manner, each candidate also computes and outputs $| I_{tA} | \in \zeta_A\{a_1^{R_AR_t}, a_2^{R_AR_t}, \ldots, a_k^{R_AR_t}\} \cap \zeta_t\{b_{t1}^{R_tR_A}, b_{t2}^{R_tR_A}, \ldots, b_{tp}^{R_tR_A}\}$, $\forall\ l = 1, \ldots, k; t = 1, \ldots, m$ and $h = 1, \ldots, p$. This output also enables each candidate to know the number of attributes s/he has in common with Alice. Hence at the end of the algorithm, each candidate will only know the number of attributes s/he has in common with Alice. Alice and Bob will get to know their actual attributes after the matched-pair has securely exchanged their random permutations.

4.3 Attacks on the Protocol and Countermeasures

In order to prevent semi-honest and malicious attacks, attributes of users of the protocol are certified. The certification binds the users′ attributes to them. As a result, the attributes cannot be modified to facilitate cheating. The certification of attributes also enhances nonspoofability of the other users′ attributes. In order to prevent collusion, it is ensured that each candidate is unaware of the presence of others in the protocol. Each candidate executes the protocol with the initiator independently. Also, as the matched-pair will eventually know the type of common attributes s/he has in common with the other user, to a large extent user profiling cannot be prevented but minimized.

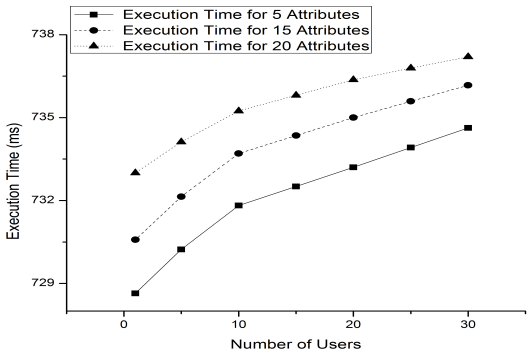


Figure 1: Comparison of execution times for the number of attributes

4.4 Experimental Implementation

In order to know the execution time, simulation for our matchmaking algorithm was conducted. The number of users with their corresponding number of attributes were the determinants of the execution time. The algorithm was simulated on an i5 PC with 2.67 GHz processor and 2G RAM. The accuracy of the execution time was ensured by taking the average of 60 repeated execution times. In the experiment, we considered varying number of users, $t = 1, 5, 10, 15, 20, 15, 30$. The initiator has the same number of attributes whilst the number of attributes each candidate has was also varied $h = 5, 15, 20$. Figure 1 shows the execution times of our algorithm for the different number of candidates with varying number of attributes.

5 Conclusion

The dynamic application of social networks has necessitated the need for secure and privacy-preserving protocols to protect users attributes from unnecessary leakage to unintended persons. In the protocol, an initiator find a candidate that has at least the threshold number of attributes without leaking any information. It can be observed that our proposed protocol for matchmaking is secure against malicious and semi-honest attacks.

References

[1] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases", in *Proceedings of SIGMOD'03*, pp. 86-97, 2003.

[2] R. W. Baldwin and W. C. Gramlich, "Cryptographic protocol for trustable match making", in *IEEE Symposium on Security and Privacy*, pp. 92-100, 1985.

[3] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy, "Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data", in *Public Key Cryptography (PKC'09)*, pp. 196-214, 2009.

[4] J. Camenisch and G. M. Zaverucha, "Private intersection of certified sets", in *Financial Cryptography and Data Security*, Springer, pp. 108-127, 2009.

[5] M. Y. Chen, C. C. Yang, and M. S. Hwang, "Privacy protection data access control", *International Journal of Network Security*, vol. 15, no. 6, pp.411-419, 2013.

[6] E. De Cristofaro, A. Durussel, and I. Aad, "Reclaiming privacy for smartphone applications", in *Proceedings of IEEE International conference on Pervasive Computing and Communications (PerCom'11)*, pp. 84-92, 2011.

[7] E. De Cristofaro, Y. Lu and G. Tsudik, "Efficient techniques for Privacy-preserving sharing of sensitive information", in *International Conference on Trust and Trustworthy Computing (TRUST'11)*, pp. 239-253, 2011.

[8] E. De Cristofaro and G. Tsudik, "Practical private set intersection protocols with linear complexity", in *Financial Cryptography and Data Security*, pp. 143-159, 2010.

[9] N. Eagle and A. Pentland, "Social serendipity: Mobilizing social software", *IEEE Pervasive Computing*, Special Issue: The Smartphone, pp. 28-34, 2005.

[10] M. J. Freedman, K. Nissim and B. Pinkas, "Efficient private matching and set intersection", in *Advances in Cryptology (EUROCRYPT'04)*, pp. 1-19, 2004.

[11] L. Kissner and D. Song, "Privacy-preserving set operations", in *Advances in Cryptology (Crypto'05)*, LNCS 3621, pp. 241-257, 2005.

[12] J. Kjeldskov and J. Paay, "Just-for-Us: A context-aware mobile information system facilitating sociality", in *Proceedings of 7th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 23-30, 2005.

[13] K. Li, T. Sohn, S. Huang, W. Griswold, "PeopleTones: A system for the detection and notification of buddy proximity on mobile phones", in *Proceedings of 6th International Conference on Mobile Systems (MobiSys'08)*, pp. 160-173, 2008.

[14] M. Li, S. Yu, N. Cao, and W. Lou, "Privacy-preserving distributed profile matching in proximity-based mobile social networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2024-2033, 2013.

[15] Y. Li, J. D. Tygar, and J. M. Hellerstein, "Computer security in the 21st Century", Chapter 3, Springer, New York, NY, USA, 2005.

[16] D. Liben-Nowelly and J. Kleinbergz, "The link prediction problem for social networks", in *Proceedings of the Twelfth Annual ACM International Conference on Information and Knowledge Management (CIKM'03)*, pp. 556-559, 2003.

[17] H. Lin, S. S. M. Chow, D. Xing, Y. Fang, and Z. Cao, Privacy preserving friend search over online social networks, *Cryptology EPrint Archive*, 2011. (http://eprint.iacr.org/ 2011/445.pdf)

[18] M. Liu and W. Lou, "FindU: Privacy-preserving personal profile matching in mobile social networks", in *Proceedings of of Infocom*, 2011.

[19] R. Lu, X. Lin, X. Liang, X. Shen, "Secure handshake with symptoms-macthing: The essential to the success of mhealthcare social network", in *Proceedings of BodyNets*, Corfu Island, Greece, 2010.

[20] R. Lu, X. Lin and X. (Sherman) Shen, "SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-health emergency", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 614-624, 2013.

[21] C. Meadows, "A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party", in *IEEE Symposium on Security and Privacy*, pp. 134-137, 1996.

[22] Y. Sang, and H. Shen, "Privacy preserving set intersection protocol secure against malicious behaviours", in *Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 461-468, 2007.

[23] S. Sarpong and C. Xu, "A secure and efficient privacy-preserving matchmaking for mobile social network", in *International Conference on Computer, Network Security and Communication Engineering, (CNSCE'14)*, pp. 362-366, 2014.

[24] A. Shamir, "Identity-based cryptosystems and signature schemes", in *Advances in Cryptology*, pp. 47-53, Springer, Berlin, Germany, 1985.

[25] J. S. Shin, and V. D. Gligor, "A new privacy-enhanced matchmaking protocol", *IEICE Transactions on Communications*, vol. E96-B, no. 8, pp. 2049-2059, 2013.

[26] J. Sun, X. Zhu, and Y. Fang, A privacy-preserving scheme for online social networks with efficient revocation, in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'10)*, pp. 1-9, 2010.

[27] Y. Wang, T. Zhang, H. Li, L. He, and J. Peng, "Efficient privacy preserving matchmaking for mobile social networking against malicious users", in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 609-615, 2012.

[28] Q. Xie, and U. Hengartner, "Privacy-preserving matchmaking for mobile social networking secure against malicious users", in *Proceedings of 9th International Conference on Privacy, Security, and Trust (PST'11)*, pp. 252-259, 2011.

[29] Z. Yang, B. Zhang, J. Dai, A. Champion, D. Xuan, and D. Li, "Esmalltalker: A distributed mobile system for social networking In physical proximity", in *IEEE ICDCS'10*, pp. 468-477, 2010.

[30] C. Zhang, J. Sun, X. Zhu, and Y. Fang, "Privacy and security for online social networks: Challenges and opportunities", IEEE Network, pp. 13-18, 2010.

[31] K. Zhang and R. Needham, "A Private Matchmaking Protocol", 2001. (http://citeseer.nj.nec.com/71955.html)

Solomon SARPONG is a Ph.D student in University of Electronic Science and Technology of China, Chengdu, (UESTC). His research interests include Information Security and Cryptography.

Chunxiang XU received her B.Sc., M.Sc. and Ph.D degrees at Xidian University, P. R. China, in 1985, 1988, 2004 respectively. She is currently engaged in Information Security, Cloud Computing Security and Cryptography as a professor at University of Electronic Science and Technology of China, Chengdu, (UESTC).

Xioajun ZHANG received his B.Sc. degree in mathematics and applied mathematics at Hebei Normal University in 2009. He also received his M.Sc. degree in pure mathematics at Guangxi University, P. R. China, in 2012. He is currently pursuing his Ph.D degree in Information Security at University of Electronic Science and Technology of China (UESTC). He is currently engaged in Cryptography, Network Security and Cloud Computing Security.

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.