IJNS
International Journal of Network Security

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

Vol. 19, No. 2 (Mar. 2017)

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. 19, No. 2 (Mar. 1, 2017)

An Investigation on Biometric Internet Security

Omprakash Kaiwartya[1], Mukesh Prasad[2], Shiv Prakash[3], Durgesh Samadhiya[2], Abdul Hanan Abdullah[1], Syed Othmawi Abd Rahman[1]

(Corresponding author: Omprakash Kaiwartya)

The Faculty of Computing, Universiti Teknologi Malaysia[1]
UTM Johor Bahru, 81310 Johor, Malaysia
National Chiao Tung University, Hsinchu, Taiwan[2]
Hsinchu 30041, Taiwan (R.O.C.)
Indian Institute of Technology, New Delhi 110016, India[3]
(Email: omprakash@utm.my)

(Received Oct. 17, 2015; revised and accepted Jan. 23 & Feb. 28, 2016)

Abstract

Due to the Internet revolution in the last decade, each and every work area of society are directly or indirectly depending on computers, highly integrated computer networks and communication systems, electronic data storage and high transfer based devices, e-commerce, e-security, e-governance, and e-business. The Internet revolution is also emerged as significant challenge due to the threats of hacking systems and individual accounts, malware, fraud and vulnerabilities of system and networks, etc. In this context, this paper explores E-Security in terms of challenges and measurements. Biometric recognition is also investigated as a key e-security solution. E-Security is precisely described to understand the concept and requirements. The major challenges of e-security; namely, threats, attacks, vulnerabilities are presented in detail. Some measurement are identified and discussed for the challenges. Biometric recognition is discussed in detail wit pros and cons of the approach as a key e-security solution. This investigation helps in clear understating of e-security challenges and possible implementation of the identified measurements for the challenges in wide area of network communications.

Keywords: Biometric, denial of service (DoS), e-network, e-security, electronic data storage, highly integrated computer-networks, malicious code threats, passive active attacks, recognition and biometric systems

1 Introduction

Information Technology has become one of the prominent support structure for any organization in this modern era. The primary goal of the technology is to provide efficient and secure flow of information flow in the organization. The present age is running on the wheels of information technology, computational devices and other value added services. Security of data has become increasingly important in any organization, and thus; organizations are working hard in their information security systems for implementing the effective and recent security approach and risk management techniques. Information security system always considers information a critical component of the organization. Protection and security of information has evolved due to unauthorized and non-authenticated changes in information of any organization.

With the rapid evolution of Internet in the last decade, e-security has become the heart of every organization. E-network based organizations can be secured with the use of modern e-security techniques. E-security provides an open and easy communication on a global platform. Network experts, administrators and data center professionals are needed to conceive the basic of security in order to carefully expand and managed today's networks. However, in recent knowledge-based economy, both the private and government organizations are using the advancements of e-revolution.

Organizations regularly observes information as an necessary resource. Data center strategic input and output has been accentuated. With the regular need of the services of e-network based systems, e-security becomes essential for almost every organizations particularly based on e-network systems. With the increasing number of computational devices connected to the network using information technology, the network security problem is becoming increasingly crowded [4].

Network security risk arises both in public as well as corporate sectors. According to the government point of view, anticipation needs to be enforced early on to counter the damage of society in terms of information security [2, 5, 26].

2 E-Security

In this section, e-security is precisely explored. E-security stands for "cyber-security", "Internet-security" as well as "IT-security".

1) E-security or electronic information security means protection of important data and information from undefined and unauthorized disclosure, transfer, modification and deletion of the information.

2) It is concerned with the security of any data that passes over the e-network in electronic form.

3) It mainly deals in securing both information as well as the network(s) through which information flows.

4) E-security is protecting an organization from internal as well as external threats and attacks.

5) It protects information networks and communication networks from the unauthorized use of the information.

6) E-security also secures the intranet, extranet from the outside world.

Due to the wide spread range, vast and continuously changing nature of communication network and environment, solutions derived e-security should be flexible, adaptable and able to detect and provide solutions to different security threats. The solutions should fulfil the requirements of the organization that are based on the e-network and information based systems.

3 Challenges for E-Security

In this section, various challenges in the design of e-security solutions are discussed. There are various challenges for e-security and various measures do exist to overcome these challenges. In spite of these measures, the challenges prevail in the form of newer vulnerabilities, threats and attacks. The security challenges are of many types and manifolds and each with dire consequences if it is not addressed properly. E-security provides open and easy communications as well as secure communication through the internet and electronic media. E-Security is a continuous process by which confidential and proprietary data and information are secured from the unauthorized and non-authenticated access from outside world in the network. Some Challenges for E-Security are listed below.

1) Protect the inadequate knowledge, data, information and tools of companies and also provides the security to internal resources and network.

2) To enable the secured exchange of confidential information, by keeping unscrupulous elements out and by providing the necessary hidden policies and methods.

3) To enable controlled access to IT network and their process, consistent with defined roles and responsibilities.

4) Make-up secure, effective, efficient, robust and trusted network for important and sensitive areas.

5) Avoid attack, fraud within business processes and transaction, and also detect the affected area also release the respond to attempted attack and fraud within the network.

6) Non-availability of e-security information knowledge is great challenges for e-security.

7) Confirm that each component of the e-world or network infrastructure is accessible when needed and also develop the verification of the transaction at the time of resource sharing.

8) To maintain trust and secure platform for transaction and processing in e-network and controlled access to computer systems and their processes, consistent with defined job and responsibilities.

Iterating all possible security factors and challenges are virtually impossible, we therefore categorized these e-security challenges in three main factors: threats, attacks and vulnerabilities. All three main factors threats, attacks and vulnerabilities are discussed in detail in the next sections.

4 Threats

Security threats are an action or process which effect on the confidentiality, integrity or availability of e-network. The threat is the result of some unauthorized disclosure, unauthorized action, and data alteration comes in the network. From an e-business concern Denial of Service (DoS) attacks come out as the most serious threat. Computer security threats are no longer limited to big companies with hundreds of employees (See Figure 1). Security threats in electronic devices such as computer deals in a various ways. There are Internet-based threats, file-based threats and social engineering threats. A security threat can be defined as potential for security violation that exists according to the circumstance, capability, action or process that could crack security and cause harm. Compute security threats could be categorized in two classes; namely, malicious code threats, transmission threats. The two threats are precisely discussed below.

4.1 Malicious Code Threats

On occasion mistakenly associated only with personal computer, malicious code can attack different platforms. Malicious code can be delivered by using various methods. For example, system penetration by the act of hacking and phasing via web sites and emails. Upon successful

delivery of malicious code, the target system can be overtaken and sensitive information can be ex-filtrated. Malicious code refers to Viruses, Worms, Trojan horses, Back door installer and other "uninvited" software. Since different name of malicious code refers to different malicious behavior, malware is common used to refer any software that performs malicious behaviors.

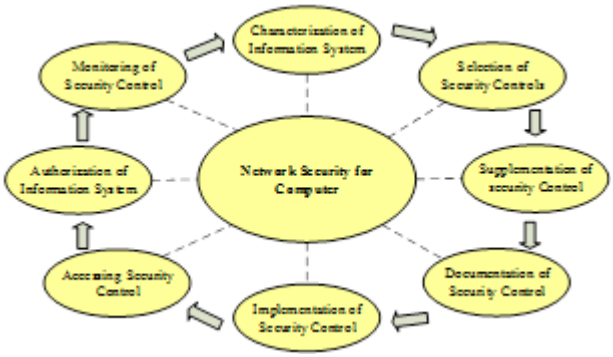


Figure 1: Network security issues

4.2 Transaction Threats

This threat happens during the huge amount of valid/invalid data is issued to a server. Because the server can only handle and process certain data threshold at any processing time or given time, the server can be overflowed with data amount that excess the data threshold vastly. As a result, the availability of the server is compromised. Generally transaction threat refers to Denial of Service (DoS), and DoS can be achieved by SYN flood, ping of death for example. Alternatively, DoS can also be achieved with large amount of legitimate requests in a short period of time

5 Attacks

Any action that makes adjustment with the security mechanism of sensitive information bought by an organization or individual is referred as attacks. The nature of attack that concerns an organization or individual varies greatly from one network to another. In general, attacks are either intentional or unintentional acts that attempt to cause of information or data loss. Ultimately all attacks are developed and originated by people with an aim to steal, cause validation, damage to the system, etc. "security attack is an attempt on system security that break up and violate the securing policy of an e-network". Security attacks are classified into two classes; namely, passive attacks, active attacks. The two attaches are precisely discussed below.

5.1 Passive Attacks

The aim of passive attack is to gain data that is being transmitted over the network. Passive attacks are in the nature of eavesdropping, and monitoring the transaction over the network. These attacks are somewhat challenging to detect because their non-involvement with any modification or alteration of information. Passive attacks can be further divided into two types:

1) Release of Message Contents: for example, eavesdropping on network transmission, detect the e-mail in the network, extracting data from the file.

2) Traffic Analysis: for example, location of data, frequency of the communication and length of the data packet.

5.2 Active Attacks

The active attacks involve with data alternation or develop the false data in the e-network. These attacks needed the attacker to be capable to transmit data to both the direction of network either may be client or server, or can block the information in one or both directions. It is also likely to perform attack like man-in-the-middle. In such scenario, the attacker can be the role of client/server during authentication procedure. The server/client will not identify that the source of the data is not the authenticated party. Active attacks may be sub-divided into four groups: Masquerade, Replay, modification message, and Denial of service.

6 Vulnerabilities

The frequent attacks during the several years, and the speed and spread of these attacks, notices a serious security vulnerability problem in our e-world. Vulnerabilities causes due to weakness of the software program or may be hardware used on a server or a client that can be exploited by a determined raider to get access to or abandon a network. We define System vulnerability as a situation, in which hacker or the attacker can access the resources and data in the absence or weakness of security constrains or technical, physical or other authorities that could be used by a threat. Vulnerabilities are flaws in a computer application that created weakness in the overall security mechanism of computer or network. Vulnerabilities in the hardware and software as well as same in policies and procedure, mainly security methods, applications and process that are using in computer networks. There is not any definite list of each and every possible sources of mentioned system vulnerabilities. The main causes of vulnerabilities are, Password management flaws, Fundamental operating system design flaws, software bugs, and unchecked user data. Some common expels of vulnerabilities are: Buffer overflows, Dangling pointers, Input validation errors, SQL injection, E-mail injection, Race condition, Simulink races, User interface failures etc.

7 E-Security Measurement And Biometric Recognition

In this section, common e-security measures have been identified and bio-metric recognition is discussed as key e-security solution.

7.1 E-Security Measurement

Effective e-security policy must consist of the objectives; namely, confidentiality, integrity, availability, legitimate use (identification, authentication, and authorization), auditing or traceability and non-repudiation. Common e-security measures are listed below.

1) Authentication: A digital certificate that approve authentication during the use of any individual's unique signing key. Basically, authentication mechanisms [6, 22] that existed today use one or more of the authenticators (factors) viz. Knowledge-based, Possession-based and Physiology-based. Knowledge-based is an authenticator only the individual knows, which typically denotes to PIN, pass phrase or a response to a secret security question. In the possession-based is an authenticator only the individual possesses, which usually refers to keys, smart cards and tokens. Physiology-based is an authenticator only the individual is or can do, referring to biometrics. Knowledge and possession-based authentication mechanisms imply that users in order to be granted access to a system, building, service - need to carry or remember the authenticator.

2) Access Controls: This limits unlike classes of users to subsets of information and make sure that they only access authorized data and services. Network constraints to safe access of other computer systems and network.

3) Encryption Policy: By this policy original data should be changed in the cipher data form for security point of view.

4) Intrusion Detection: These product monitor system and network activity to spot any attempt being made to gain access.

A major e-security policy of is ISO/IEC 27001. The main features of ISO/IEC 27001 are; namely, reviewing, the security policy, acceptable use policy. The prevention methods can be also applied for malware, such as antivirus software, firewalls deployment, preventing the download of extruder programs and documents by the Internet and make sure your staff adhere to this policy and also apply the malware alert service in computer network and system. The key components of fine-grained access control implementation in a corporate server environment. The above analysis of e-security and its measures gives some precise practices for e-security which are described in next paragraph.

Either defines, deploys organization level security policies or enabled security threat preventive measures. Check the authorization level of computer network from the outside world. With appropriate modification, these measures can also be applied in wide area of communication networks where security is prime concern during information forwarding in communication [6, 10, 11, 12, 13, 14, 15, 16, 17, 19, 32, 33].

7.2 Biometric Recognition: As a Key E-Security Solution

Biometric Recognition is the statistical analysis and measurement of folks' physical and interactive characteristics [21, 23]. The technique is primarily used for identification, access control and identifying individuals that are under surveillance. The basic evidence of biometric authentication is that each person is unique. She can be identified by her intrinsic physical or behavioral traits. The term "biometric" is consequent from the Greek words "bio" means life and "metric" means to measure. Information security is an unlimited concern to electronic users, which is not only a technical challenge, but also related to human factors. For example one of the key issues in Malaysia related to Internet banking is the pathetic security used for Internet banking application. Hence it is an important study to investigate further the solution and enhance the security issues in Internet banking applications. Modern world is fast world, billions of transactions occur each minute. On behalf of these transactions, data prerequisite to be readily available for the unpretentious. Biometric Recognition is normally considered as physiological or behavioral characteristic based recognition [3]. Physiological characteristic represents to stable characteristics of human beings including fingerprints, structure of face, eyes, ears, hands, legs, and fingers, the pattern of hairs, teeth, and samples of DNA structure. Physiological characteristics of each individual are normally permanent and distinctive. It changes only due to fatal accidents, serious illnesses, genetically induced defects, or in some cases changes due to aging. Behavioral characteristics are represented by the day-to-day way of living life by the particular human being. Interactions with other human beings are also include to measure the behavioral characteristics. Biometric recognition can be utilized as a key e-security solution. Biometric recognition generally works in five steps. The steps and the actions needs to be executed in the corresponding steps are listed below.

1) Sample acquisition: Collection of biometric data using appropriate sensors.

2) Feature extraction: Conversion of biometric data into templates.

3) Storage: Storage of templates in appropriate memory which depends on the application.

4) Matching: authentication of user by comparing bio-

metric template of the user with the existing templates stored in the database.

5) Decision: Based on the result of the matching, the user will be authorized or denied to access the resources.

Biometric system is a pattern recognition system that recognizes the user's identity through their physical or behavioral traits. For using biometric system authentication, the user has to firstly enrollment in biometric systems. Based on application context, biometric system operates in one of two modes; verification and identification for user authentication. During enrollment (cf. Figure 2), features of the individual are extracted from the sensor or user interface and converted into templates. The templates are stored in the system database. Verification (cf. Figure 3) is used for one to one match. It validates a person's identity by comparing his captured biometric traits with the specific biometric template that is stored in system database. Verification method can be used with either centralized storage or distributed storage.

In verification with centralized storage, database exists in a centralized mode in which all biometric templates are stored. From the system database, a specific template is retrieved for comparison with live person's physical or behavioral traits and both data can be either match or not. Two types of error is possible in verification; namely, false match or false positive, false negative or false rejection. In false matches or false positive, a person is not who he claims, but the system accepts it, Acceptance of pretender. In false rejection or false negative, a person is who he claims to be but the system fails to accept, rejection of legitimate person. False rejection will cause unnecessary inconvenience to an innocent individual, whereas the false match is more dangerous as they allow an imposter to pass.

In verification with distributed storage, biometric data stored in a memory device that is carried by individual, for example, smart card in which biometric data of individuals are stored. Verification is done by comparing biometric data which are provided by an individual's memory device like smart card with a specific template stored in the system database. Like, verification with centralized storage, false acceptance and false rejection errors are possible Verification with distributed storage technique. Unlike, verification with centralized storage, memory device like token or smart card can be damaged or can be tampered.

Identification (cf. Figure 4) is used for one to many matches in biometric recognition process. In particular, it is used to discover the identity of a person when the identity is unknown; in this mode biometric data of the individual is captured and compared with all templates which stored in the system database. To determine the identity of the unknown person, database of templates is required that contains biometric templates of all people known to the system. Identification not possible without database of templates. Like verification, identifica-

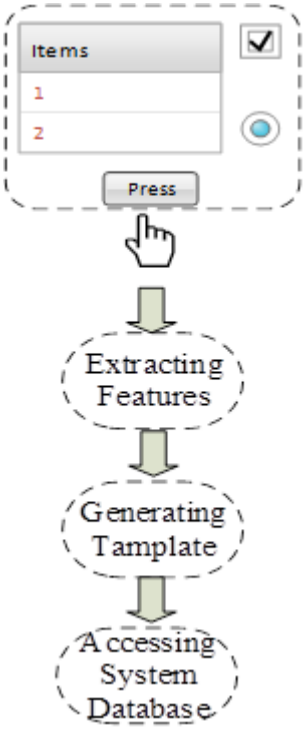


Figure 2: Enrollment in biometric recognition

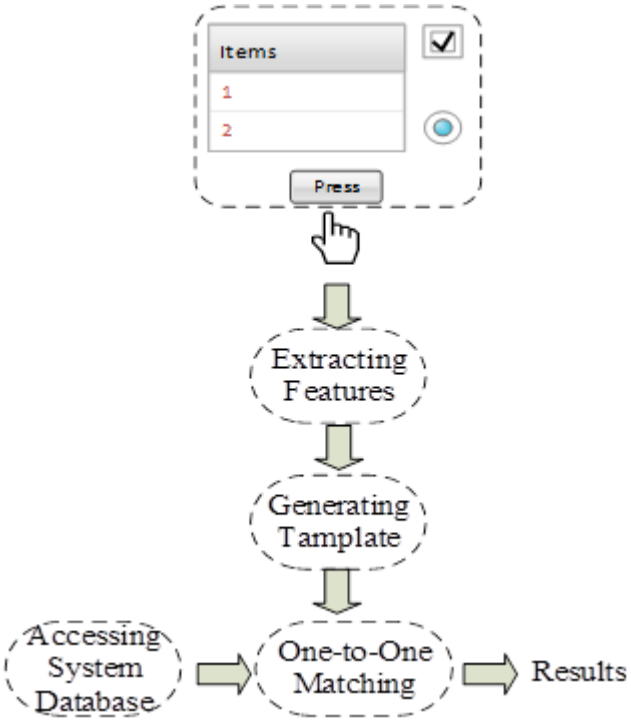


Figure 3: Verification in biometric recognition

tion can also produce two types of error, false match or false reject. There are two types of biometrics; namely, physical traits, behavioral traits (cf. Figure 5). physical biometrics includes five factors of physical attributes that can be used for user authentication. The five factors are discussed below.

1) Fingerprint scan: It is largely watched as an precise biometric recognition method. Nowadays, fingerprint scanners are available at low cost and progressively integrated in electronic devices [1]. Among all the biometric techniques, this is the oldest method which has been successfully used in numerous applications. For example, fingerprint scan use in forensic for criminal identification, use in attendance system. Because the patterns of ridges on the fleshy part of fingertips are unique. No two individuals even twins have same fingerprints. The patterns of ridges leave impression on whatever they touch. Injuries such as minor burns or cuts do not mop out or change the pattern , the new skin grows showing the same pattern.

2) Retina or iris scan: The idea of distinguishing an individual by using iris patterns was suggested by an ophthalmologist in 1936. Later, the idea appeared in some action movies, including 1983's James Bond "Never Say Never Again", nonetheless at that time it remained science fiction. In 1994, the first automated iris pattern recognition algorithm was proposed by physicist and computer-vision expert John Daugman and patented, and continue to be the basis of all current iris recognition systems and products. These have been used to confirm a person's identity by reading the arrangement of blood vessels in the retina or patterns of color in the iris. It is very reliable technique and difficult to map by forgers.

3) Facial recognition: This technique use unique facial attributes to identify an individual. Biometric facial recognition systems generally read the overall structure, shape and proportions of the face taking into account the distance between the eyes, nose, mouth, and jaw edges, upper outlines of the eye sockets, the sides of the mouth, the location of the nose and eyes, etc.

4) Finger vein recognition: The technology of finger vein recognition is quite younger then fingerprint or facial recognition system. Finger vein recognition system uses pattern-recognition techniques based on images of human finger vein patterns under the skin's surface. Key advantage of vein patterns for biometric identification is that the forgers cannot easily create a copy of finger vein due to lack of known method, as it is possible with fingerprints.

5) Palm vein recognition: As the blood vessels zigzag beneath our skin these give a unique pattern that can be used to identify a person. Infrared beam is used to enter in the hand and the veins in the person's palm show up as black lines .Like the finger vein, Palm vein also cannot be easily forged. Due to that they can provide highest level of security.

DNA based recognition: At this time, there exists no technique to allow for immediate and automated recognition of DNA samples. DNA analysis and profiling (genetic fingerprinting) needs a lab environment for a number of hours. Though, significant research and development efforts are ongoing to develop this technique, and also to enable governments to better use the millions of DNA profiles collected and archived in databases of DNA.

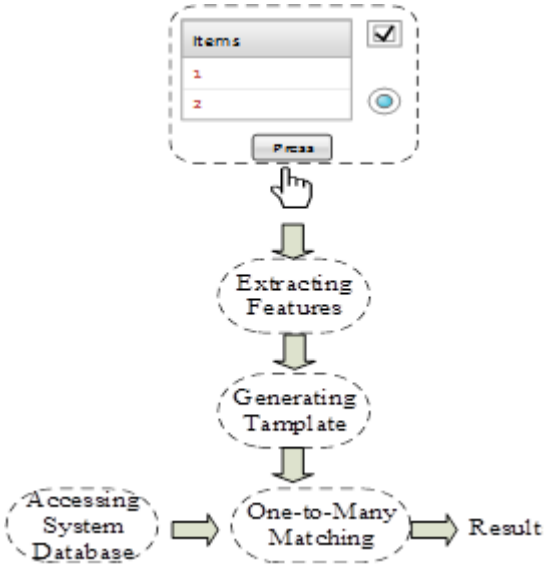


Figure 4: Identification in biometric recognition process

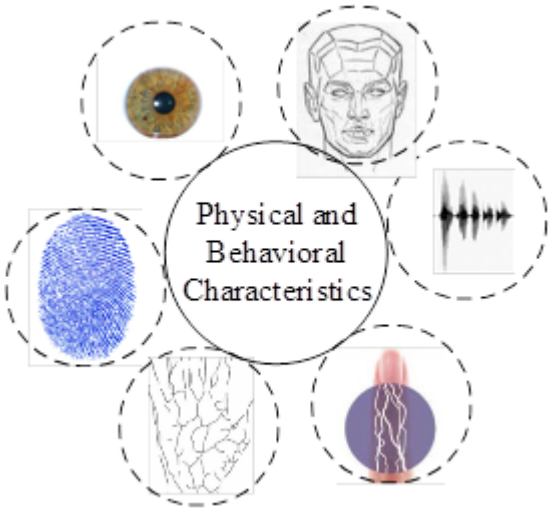


Figure 5: Physical and behavioral characteristics in biometric recognition

Behavioral biometrics includes three factors of physical attributes that can be used for user authentication;

namely, voice, signature and keystroke traits. The three factors are discussed below.

1) Voice scan: It is behavioral biometric which can be learned or acquired, but also include physiological elements. E.g., the human voice is influenced by the physiological characteristics of lungs, tongue, throat, etc. and its behavioral features evolve and change over time. They can be influenced by factors such as age, illnesses, mood, conversational partner or surrounding noise. It uses a voice print that analyses how a person says a particular word or sequence of words unique to that individual.

2) Signature scan: Signature scan is a process of, investigating an individual's signature. The technology investigates speed, direction, and pressure of writing, total time of the signature. Unluckily, signature is one of the least reliable methods of Identification. Forgers have number of ways to reproduce a signature that looks similar to the owner. Signature dynamics is biometric signature recognition systems measure and analyze the physical activity of signing. Important features include stroke order, the pressure applied, the pen-up movements, the angle the pen is held, the time taken to sign, the velocity and acceleration of the signature. Some systems moreover compare the visual image of signatures; however the focus in signature biometrics lies on writer-specific information rather than visual handwritten content.

3) Keystroke scan: It is the recognition of keystroke dynamics is the process of investigating the way an individual types at a terminal by monitoring the keyboard inputs thousands of times per second in an attempt to recognize the individual based on habitual typing rhythm patterns. Keystroke dynamics are described by speed (the time a key is pressed, the time between keys pressed), rhythm, precision, keys used (e.g., left Shift key or right Shift key, Caps Lock), and other typing features.

The major issue of biometric recognition is that the information used in biometric recognition changes with age and occupation of persons. Also, It may change or damage due to suffering from physical injuries or diseases. Like, password or card biometric data cannot be changed after any misshaping. Advantage of biometric data is that, they cannot get stolen, lost, replicated or forgotten like password or token, cards. They also can not be forgotten, compromised, shared, observed or guessed like password, secret codes or PIN. You don't need to change biometric data from time to time as you do with passwords. No need to write them as most people write password. Due to high security level and accuracy of biometric authentication, government agencies are also attracted towards the biometric authentication method, for example Indian government used this technology to recognize country's people and name of this program is adhaar. In modern world, improved performance and availability of equipment at low cost and automated biometric recognition. Therefore biometric applications are categorized into three key sets which are as follows

1) Forensic applications: These are used in criminal investigations and archeology. For example, corpse identification, parenthood determination, identifying historical fact from fossils, etc.

2) Government applications: This category includes personal documents, for example voter id, passports, ID cards, etc.

3) Commercial applications: This category includes physical access control; network logins; e-Commerce, i-Commerce, m-Commerce, ATMs; credit cards; device access to computers, mobile phones; e-Health etc.

A number of provisions and techniques [36] have been suggested to safeguard security and privacy in biometrics. These are Multimodal biometric, Template-on-token, Match-on-token and Data-hiding methods attack. Data-hiding techniques embed additional information in fingerprint images - an approach similar to hiding digital watermarks in image or audio data to ensure data integrity. If the embedding algorithm remains secret, a service provider can investigate the received fingerprint image for the expected standard watermark to ensure it has been sent from a trusted sensor.

Biometric Systems Development: It is a combination hardware/software system for biometric identification or verification. Key functions of a biometric system are

1) Receive biometric samples from an individual.

2) Extract biometric feature from the sample.

3) Compare the sample of the candidate with stored templates from individuals.

4) Indicate identification or verification upon the result of the previous comparison.

Biometric systems have components which are an automated mechanism and interface with application systems. These pieces may be configured to suit different situations. A common issue is where the stored images reside; on a card presented by the person being verified or at host computer. Recognition occurs when an individual's is matched with one of a group of stored images. Biometric accuracy is the system's ability of separating legitimate matches from imposters. Although biometrics technology provides a strong factors. In [27] security priorities scheduling in Network is discussed with focus on security optimization. The scheduling which considered security, all task want execute in secure mode. The most important Network users' requirements can be representing in form many arrangements therefore it falls under category of NP Class Problem [7, 8, 18, 20, 25, 28, 29, 30, 31, 34, 35]. Then GA and variants applied for single objective optimization. Then NSGA II and variants applied [24] for multi-objective optimization.

8 Conclusions

Because of the changing security maps, clearly some organizations are required to arrange security solutions that preserve adequate counter-measures for every threat and provide the capability to meet industry regulations- Security is very big social issue as a practical and technical one. According to the new economy, information is important both as input and output. Hence information security measurement must be high priority. Security needs must constantly keep pace with ever changing technologies and application. The e-security challenges are many both from the viewpoint of their categories as well as from the view of their way of information implementation in the network.

E-security cannot be achieved through technology alone. There is almost boundless number of ways by which e-network or e-world set up could be assaulted by hackers, crackers and disgruntled insiders. Common threats include active and passive attacks, hacking, malware, Denial of Service (DoS), and vulnerabilities. We control the e-security issue by the three basic attributes known as C.I.A. (Confidentiality, Integrity, and Availability). India has taken a number of strategic initiatives to strengthen information security. This has included the enactment of the Information Technology Act 2000. ISO/IEC 27001 provides a sound basis for the development of a security policy. Biometric recognition is studied and being implemented as a key approach for e-security solution.

Acknowledgments

The research is supported by Ministry of Education Malaysia (MOE) and conducted in collaboration with Research Management Center (RMC) at University Teknologi Malaysia (UTM) under VOT NUMBER: Q.J130000.2628.11J31. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. H. Aghdam, P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *International Journal of Network Security*, vol. 18, no. 3, pp. 420–432, 2016.

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

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

[4] G. G. Deverajan, R. Saravanan, "A novel trust based system to detect the intrusive behavior in MANET," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 31–43, 2015.

[5] W. R. Ghanem, M. Shokir, and M. Dessoky, "Defense against selfish PUEA in cognitive radio networks based on hash message authentication code," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 12–21, 2016.

[6] K. Gupta, K. Gupta, O. Kaiwartya, "Dynamic ad hoc transport protocol (D-ATP) for mobile Ad hoc networks," *IEEE 5th International Conference on Generation Information Technology Summit (Confluence)*, pp. 411–415, 2014.

[7] D. He, J. Chen, and J. Hu, "Weaknesses of a remote user password authentication scheme using smart card," *International Journal of Network Security*, vol. 13, no. 1, pp. 58–60, 2011.

[8] B. Kadri, M. Feham, and A. Mhammed, "Efficient and secured ant routing algorithm for wireless sensor networks," *International Journal of Net-work Security*, vol. 16, no. 2, pp. 149–156, 2014.

[9] B. Kadri, M. Feham, and A. Mhammed, "Efficient and secured ant routing algorithm for wireless sensor networks," *International Journal of Network Security*, vol. 16, no. 2, pp. 149–156, 2014.

[10] O. Kaiwartya, S. Kumar, "Geocast routing: Recent advances and future challenges in vehicular adhoc networks," in *IEEE International Conference on Signal Processing and Integrated Networks (SPIN'14)*, pp. 291–296, 2014.

[11] O. Kaiwartya, S. Kumar, "Cache agent based geocasting (CAG) in VANETs," *International Journal of Information and Communication Technology*, vol. 7, no. 6, pp. 562–584, 2015.

[12] O. Kaiwartya, S. Kumar, "Enhanced caching for geocast routing in vehicular Ad hoc network," in *Intelligent Computing, Networking, and Informatics*, pp. 213–220, Springer, 2014.

[13] O. Kaiwartya, S. Kumar, "Geocasting in vehicular adhoc networks using particle swarm optimization," in *ACM Proceedings of the International Conference on Information Systems and Design of Communication*, pp. 62–66, 2014.

[14] O. Kaiwartya, S. Kumar, "Guaranteed geocast routing protocol for vehicular adhoc networks in highway traffic environment," *Wireless Personal Communications*, vol. 83, no. 4, pp. 2657-2682, 2015.

[15] O. Kaiwartya, S. Kumar, R. Kasana, "Traffic light based time stable geocast (T-TSG) routing for urban VANETs," in *IEEE Sixth International Conference on Contemporary Computing (IC3)*, pp. 113–117, 2013.

[16] O. Kaiwartya, S. Kumar, D. K. Lobiyal, A. H. Abdullah, A. N. Hassan, "Performance improvement in geographic routing for vehicular Ad hoc networks," *Sensors*, vol. 14, no. 12, pp. 22342–22371, 2014.

[17] O. Kaiwartya, S. Kumar, D. K. Lobiyal, P. K. Tiwari, A. H. Abdullah, A. N. Hassan, "Multiobjective dynamic vehicle routing problem and time seed based

solution using particle swarm optimization," *Journal of Sensors*, vol. 2015, Article ID 189832, 14 pages, 2015.
[18] M. Kumar, "An enhanced remote user authentication scheme with smart card," *International Journal of Network Security*, vol. 10, no. 3, pp. 175–184, 2010.
[19] R. Kumar, S. Kumar, D. Shukla, R. S. Raw, O. Kaiwartya, "Geometrical localization algorithm for three dimensional wireless sensor networks," *Wireless Personal Communications*, vol. 79, no. 1, pp. 249–264, 2014.
[20] 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.
[21] 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, Jan. 2013.
[22] C. T. Li, M. S. Hwang, "An online biometrics-based secret sharing scheme for multiparty cryptosystem using smart cards," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 5, pp. 2181–2188, May 2010.
[23] C. T. Li and M. S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards", *Journal of Network and Computer Applications*, vol. 33, pp. 1–5, 2010.
[24] 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.
[25] C. Lin, Y. Li, K. Lv, and C. C. Chang, "Ciphertext-auditable identity-based encryption," *International Journal of Network Security*, vol. 17, no. 1, pp. 23–28, 2015.
[26] E. U. Opara, O. A. Soluade, "Straddling the next cyber frontier: The empirical analysis on network security, exploits, and vulnerabilities," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 10–18, 2015.
[27] S. Prakash and D. P. Vidyarthi, "Observations on effect of IPC in GA based scheduling on computational grid," *International Journal of Grid and High Performance Computing*, vol. 4 no. 1, pp. 66–79, 2012.
[28] S. Prakash and D. P. Vidyarthi, "Load balancing in computational grid using genetic algorithm," *International Journal of Advances in Computing, Scientific and Academic Publishing*, vol. 1 no. 1, pp. 8–17, 2011.
[29] S. Prakash and D. P. Vidyarthi, "Immune genetic algorithm for scheduling in computational grid," *Journal of Bio-Inspired Computing*, vol. 6, no. 6, pp. 397–408, 2014.
[30] S. Prakash and D. P. Vidyarthi, "A novel scheduling model for computational grid using quantum genetic algorithm," *Journal of Supercomputing*, vol. 65, no. 2, pp. 742–770, 2013.
[31] S. Prakash and D. P. Vidyarthi, "Maximizing availability for task scheduling in computational grid using GA," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 1, pp. 197–210, 2015.
[32] M. Prasad, K. P. Chou, A. Saxena, O. P. Kawrtiya, D. L. Li, C. T. Lin, "Collaborative fuzzy rule learning for Mamdani type fuzzy inference system with mapping of cluster centers," in *IEEE Symposium on Computational Intelligence in Control and Automation (CICA'14)*, pp. 1–6, 2014.
[33] R. S. Rao, S. K. Soni, N. Singh, O. Kaiwartya, "A probabilistic analysis of path duration using routing protocol in VANETs," *International Journal of Vehicular Technology*, vol. 2014, Article ID 495036, 10 pages, 2014.
[34] C. Y. Sun and C. C. Chang, "Cryptanalysis of a secure and efficient authentication scheme for access control in mobile pay-TV systems," *International Journal of Network Security*, vol. 18, no. 3, pp. 594–596, 2016.
[35] J. Wei, W. Liu, X. Hu, "Secure andefficient smart card based remote user password authentication scheme," *International Journal of Network Security*, vol. 18, no. 4, pp. 782–792, 2016.
[36] H. Zhu, Y. Zhang and X. Wang, "A novel one-time identity-password authenticated scheme based on biometrics for e-coupon system," *International Journal of Network Security*, vol. 18, no. 3, pp. 401–409, 2016.

Omprakash Kaiwartya received his Ph.D. and M.Tech degrees in Computer Science from School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India in 2015 and 2012 respectively. He is currently associated with Faculty of Computing, Universiti Teknologi Malaysia (UTM), Skudai Johor, Malaysia as Post-Doctoral Fellow. His research interests include Vehicular Ad-hoc Networks, Mobile Ad-hoc Networks and Wireless Sensor Networks. Dr. Omprakash has published papers in International Journals and Conferences with publishers including ACM, IEEE, Elsevier, Springer, MDPI, KIIS, InderScience and Hindawi.

Mukesh Prasad (M'13) received his Ph.D. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan in 2015 and master degree in computer application from Jawaharlal Nehru University, New Delhi, India, in 2009. He is currently associated with National Chiao Tung University, Hsinchu, Taiwan as Post-Doctoral Researcher. His current research interests include machine learning, pattern recognition, fuzzy systems, neural networks and Wireless Sensor Networks. Dr. Mukesh has published papers in international journal papers and conferences including IEEE Transactions, ACM and Springer.

Abdul Hanan Abdullah received his Ph.D. degree from Aston University in Birmingham, United Kingdom in 1995. He is currently working as a Professor at Universiti Teknologi Malaysia (UTM). He was the dean

at the Faculty of Computing, UTM from 2004 to 2011. Currently he is heading Pervasive Computing Research Group, a research group under K-Economy Research Alliances. His research interests include wireless sensor networks, Internet of Things, network security and next generation networks. Prof. Abdullah has published papers in International Journals and Conferences including IEEE, Elsevier, Wiley & Sons, Springer, MDPI and Hindawi.

Shiv Prakash received his Ph.D. and M.Tech degrees in Computer Science from School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi, India in 2014 and 2010. His research interest includes parallel/distributed system, grid computing, Cloud computing, machine learning. He has published papers in various international journals and peer-reviewed Conferences including IEEE, ACM, Elsevier, Springer, Wiley & Sons and Inderscience.

Durgesh Samadhiya is a scientist at the Emerging Device Division department of National Applied Research Laboratories, Hsinchu Taiwan, working on the IoT tag for all the products around the world that can give you the information about the product and location of the product. Previously he was a Research Fellow at Chung Hua University, Taiwan. He holds a PhD degree in Method Engineering of Software Engineering from Taiwan, Master Degree in computer Science from College of Engineering Roorkee, India. He has published good number of research papers in International Journal and Conference proceeding indexed in Science Citation Index, Ei compendex, Google Scholar, database and logic programming (DBLP), Web of Science.

Syed Othmawi Abd Rahman is currently working working at Faculty of Computing, Universiti Teknologi Malaysia (UTM). He has received post graduate diploma from University of Warwick, UK in 1990 and master degree from UTM in 1996.

A Secure and Efficient One-time Password Authentication Scheme for WSN

Chung-Huei Ling[1], Cheng-Chi Lee[2], Chou-Chen Yang[3], and Min-Shiang Hwang[1,4]
(Corresponding author: Min-Shiang Hwang)

Department of Computer Science and Information Engineering, Asia University[1]
No. 500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
(Email: mshwang@asia.edu.tw)
Department of Library and Information Science, Fu Jen Catholic University[2]
Department of Management Information Systems, National Chung Hsing University[3]
Department of Medical Research, China Medical University Hospital, China Medical University[4]
No. 91, Hsueh-Shih Road, Taichung 40402, Taiwan
(Received May 17, 2015; revised and accepted Aug. 21 & Sept. 8, 2015)

Abstract

An algorithm that a user has authenticated over remote devices should be designed to consider the limitations of computation and lower power in a wireless sensor networks. Lamport first proposed a one-time password authentication scheme which the password was different in each transaction. In this paper, according to the Lamport's concept we propose an efficient and secure one-time password authentication scheme for wireless sensor networks.

Keywords: Authentication, one-time password, security, wireless sensor networks

1 Introduction

Now, people take up a variety of actions through public network, such as shopping, business transaction, obtaining new information, etc. In consideration of those requirements, there are more service providers to supply the services. In order to avoid people misusing the resources, the provider should ensure the user's valid identity, and limit the user's rights of uses [8].

One of the simplest user authentications over insecure networks is password authentications [19]. It allows the legal users to use the resources of the remote systems via internet. However, it is vulnerable to various attacks in internet environment, such as guessing attack, replay attack, modification attack, and stolen verifier attack, etc. Therefore, Lamport firstly proposed a one-time password authentication concept [10], which requires different verified information such as password in every transaction. After that, a number of researchers have proposed several password authentication schemes for secure login of legal users [1, 11, 12, 13, 15, 22].

There are many applications in Wireless Sensor Networks (WSN): military sensing, wild animal tracking, environment monitoring, health monitoring, etc. [2, 4, 5, 16, 17, 18, 20, 23, 26]. The security issue is always an important in WSN [9, 21]. In 2004, Watro et al. proposed a tiny public key technology for securing WSN [24]. In 2006, Wong et al. proposed a dynamic user authentication scheme for WSNs [25]. Their schemes were simple and efficient to implement. However, Das pointed out that their schemes are insecure [3]. Das also proposed a two-factor user authentication in WSNs in 2009. However, Lee et al. shown that Dan's scheme is also insecure against masquerade attack in 2011 [14]. In this paper we will propose an efficient and secure one-time password authentication scheme for WSN.

2 The Proposed Scheme

In this paper, we propose a secure and efficient one-time password authentication scheme for WSN. First, we briefly summarized our idea. In order to reduce the computation of the mobile devices, the method of the hash chain is removed, but still retains the one-way hash function to achieve mutual authentication. There is no hash chain in our scheme, so the user cannot consider the login times. The property will make the authenticated algorithm more flexible for the user. Three participants are in the proposed scheme: the login users, gateway node (GW_{Node}), and sensor node (S-node). In this scheme, each user holds his/her user's identity (ID) and password. Each user uses his/her ID and password to login to the GW_{Node} with smart card.

There are three phases in the proposed scheme: the registration, login and authentication, and password change phases. To be a legal user, each new user has

Table 1: The notations used in the proposed scheme

Notations	Description
U_i	User i.
S_n	Sensor node's identity.
GW_{Node}	Gateway node of WSN.
S_{Node}	Sensor node of WSN.
$SEED$	A random number which is chosen by the GW_{Node} and stored in some designated S_{Nodes}.
T, T'	Timestamp.
D	A random number.
K	User's secret value.
p_0	Initial password of U_i.
$H(\cdot)$	Cryptographic one-way hash function.
$H^2(\cdot)$	Hashing twice using cryptographic one-way hash function.
$\|\|$	Concatenation of bits.
$\oplus$	XOR operation.
$A \rightarrow B : message$	The messages are transmitted from A to B.

to register with the GW_{Node} of WSN in the registration phase. After this phase, the new user could obtain a valid user identity (ID), password, and a smart card from the GW_{Node}. After that, the legal user could login to the S_{Node} and GW_{Node} with his/her ID, password, and smart card in the login phase. Next, the GW_{Node} validates the user legitimacy in the authentication phase. If the user passes the validation, the user could have a privilege to access data and service from the GW_{Node}. Once the users want to change his/her password, they can use the password change phase to change their passwords. The detailed process is described in the following. The abbreviations and notations used throughout the paper are shown in Table 1.

A. Registration Phase

There are three steps in registration phase and they are illustrated as follows.

$User \leftarrow GW_{Node} : smartcard(SEED);$

$User \leftarrow GW_{Node} : T, SEED \oplus D, H(D||T);$

$User \rightarrow GW_{Node} : p_0 \oplus D', H(p_0).$

Before registering, the user will receive a smart card which contains a pre-shared secret value $SEED$, where the $SEED$ is a random number which is chosen by the GW_{Node}, and the GW_{Node} keeps the $SEED$. As the GW_{Node} receives a user's registered request, the GW_{Node} will transmit a timestamp T, $SEED \oplus D$, and a hash of $(D||T)$ to the user, where D is a random number. When the user receives this information, he/she extracts D' from the equation of $(SEED \oplus D) \oplus SEED$. Then, the user computes $H(D'||T)$, and compares with $H(D||T)$. If the two values are equal, the user can verify the identity of the GW_{Node}.

After verifying the GW_{Node}'s identity, the user computes the initial password $p_0 = H^2(K \oplus SEED)$, and then protects p_0 against being modified with hash function such as $H(p_0)$, where K is the user's secret value. The user transmits the $p_0 \oplus D'$, and $H(p_0)$ to the GW_{Node}. When the GW_{Node} receives the messages, he/she extracts p_0 from the equation $(p_0 \oplus D') \oplus D$. The GW_{Node} computes and compares with the received $H(p_0)$. If the two values are equal, the GW_{Node} can verify the user's identity. Finally, he/she stores the initial password p_0.

B. Login and Authentication Phase

There are two steps in login and authentication phase and they are illustrated as follows.

$User \leftarrow GW_{Node} : T, SEED \oplus D_t, H(D_t||T);$

$User \rightarrow GW_{Node} : T', p_t.$

For the login, the GW_{Node} sends the timestamp T, $SEED \oplus D_t$, and $H(D_t||T)$ to the user, where D_t is a random number used in this transaction. As the user receives the messages, he/she will perform the same procedures as the registration phase. First, he/she checks the timestamp T is the current time or not. If not, he/she rejects the transaction and responds to the failure of the GW_{Node}. Next, he/she extracts D'_t from $(SEED \oplus D_t) \oplus SEED$, and then compares the $H(D'_t||T)$ with $H(D_t||T)$. If the two values are equal, he/she can verify the GW_{Node}'s identity. Afterward the user computes a verified value p_t as follows: $p_t = H(K \oplus SEED) \oplus H(p_0||T'||D'_t)$, and sends the p_t with timestamp T' to the GW_{Node}. When the GW_{Node} receives the two values, he/she checks the timestamp and computes the value $x = H(p_0||T'||D_t) \oplus p_t$. Then, the GW_{Node} computes the hash of the value x suchlike $H(x)$, and verifies the equation $H(x) \stackrel{?}{=} p_0$. If the equation is equal, the GW_{Node} could verify the user's ID.

Next, the GW_{Node} computes $A_i = h(S_n \,||\, SEED \,||$

T'), where T' is the current timestamp of GW_{Node}; S_n is an identity of S_{Node} . The GW_{Node} sends $\{A_i, T'\}$ to the S_{Node} S_n. The S_n then verifies the timestamp T' and computes $A'_i = h(S_n||SEED||T')$. If A_i is equal to A'_i and the timestamp is correct, the S_n will respond to U_i's query.

C. Password Change Phase

There are two steps in password change phase and they are illustrated as follows.

$User \rightarrow GW_{Node} : SEED \oplus p'_0, H(p_0||p'_0)$;
$User \leftarrow GW_{Node} : H(p'_0||1)$.

If U_i wants to change his/her password, the following procedure is performed.

U_i selects a new secret value K' and then calculate $p'_0 = H^2(K' \oplus SEED)$.

U_i calculates $SEED \oplus p'_0$ and $H(p_0||p'_0)$.

U_i sends $SEED \oplus p'_0$ and $H(p_0||p'_0)$ to the GW_{Node}.

GW_{Node} calculates $p'_0 = SEED \oplus p'_0 \oplus SEED$ and $H(p_0||p'_0)$.

GW_{Node} checks the computed value $H(p_0||p'_0)$ is equal to the received value $H(p_0||p'_0)$.

If they hold, GW_{Node} stores the p'_0 in place of p_0 and calculates $H(p'_0||1)$.

GW_{Node} sends $H(p'_0||1)$ to the user to ensure that he/she changes his/her new password successfully.

In our scheme, we utilize the one-way hash function, timestamp, and a random number to achieve the requirements of a one-time password authentication scheme, which there are different verification values in each transaction.

3 Security Analysis

In this section, we consider the variable possible attacks in the design of a one-time password authentication scheme for WSN, such as server spoofing attacks, stolen-verifier attacks, pre-play attacks, active attacks and revealing message contents, off-line dictionary attacks, and replay attacks, etc.

Server Spoofing Attack Analysis:

There is a malicious attacker masquerading as a GW_{Node} to obtain some secret information about the user. As the user cannot detect the kind of spoofing attack, he/she will reveal his/her secret message by accident. Our scheme can prevent this kind of attack. The GW_{Node} will be authenticated by the pre-shared secret value SEED. If an attacker wants to replay the authentication message, he/she must modify the timestamp T to T'. However, it is not easy that an attacker knows the random number D, and then computes another hash value of D and T'. Therefore, the attacker cannot pass the authentication.

Stolen-Verifier Attack Analysis:

Assume that an attacker has stolen the password-verifier, $p_0 = H^2(K \oplus SEED)$, from the GW_{Node}. He/she cannot recover $H(K \oplus SEED)$ from $H^2(K \oplus SEED)$ since $H(\cdot)$ is a strong one-way hash function [3, 11]. Therefore, the user who knows the password K can compute the value $H(K \oplus SEED)$ and then pass the authentication.

Pre-play Attack Analysis:

If a challenge is predictable in a challenge-response protocol, the possible attack of "suppress-replay attack" will happen. In our scheme, it is almost impossible that an attacker forecasts the next random number D and then modifies the timestamp T to a legitimate time. Therefore, the attacker cannot forge a valid user to login the GW_{Node} and request services.

Off-line Dictionary Attack Analysis:

Generally, users always select a secret key which is easy to remember and guess. The secret key will easily suffer from guessing attacks, especially off-line dictionary guessing attacks. Herein, we used a large random number $SEED$ to protect the user's secret key K. The attacker cannot guess the correct password K and $SEED$ simultaneously, and he/she finds it hard to obtain the secret key K.

Active Attack and Revelation of Message Contents Analysis:

According to RFC1704 [6], active attack is when an attacker attempts to modify data improperly, gain authentication, or gain authorization by modifying transmitted messages. In order to maintain the integrity and the confidentiality of transmitted messages, the sender and the receiver should encrypt the messages. In our scheme, we establish a session key D to encrypt the transmitted messages in each communication.

Replay Attack Analysis:

An adversary eavesdrops on the valid user's verified information T' and p_t. If he/she replays the message to forge the valid user, he/she will be rejected. Because a timestamp is contained in p_t, the GW_{Node} can check it and reject the adversary's request.

Portability Analysis:

Smart card has a property which can be portable. In S/Key [7], the fresh one-time passwords should be pre-computed by the user. On a trip where no trusted local computation is available, the user can use the pre-computed password to login the server. It is not convenient for a user that he/she should pre-compute the next one-time passwords. So, we continue using the smart card to keep the portability of the algorithm.

4 Conclusion

In this paper, we have proposed an efficient and secure one-time password authentication scheme for WSN. The proposed scheme is secure to against Lee et al.'s masquerade attacks [14], pre-play attacks, the server's spoofing attacks, active and revelation of message contents attacks, off-line dictionary attacks, stolen-verifier attacks, and replay attacks.

Acknowledgements

This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: MOST 103-2221-E-468 -026, NSC 103-2622-E-468-001-CC2, and NSC 103-2622-H-468-001-CC2.

References

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

[2] M. Asadi, C. Zimmerman, and A. Agah, "A game-theoretic approach to security and power conservation in wireless sensor networks," *International Journal of Network Security*, vol. 15, no. 1, pp. 50–58, 2013.

[3] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[4] T. H. Feng, W. T. Li, and M. S. Hwang, "A false data report filtering scheme in wireless sensor networks: A sSurvey," *International Journal of Network Security*, vol. 17, no. 3, pp. 229–236, 2015.

[5] T. H. Feng, N. Y. Shih, and M. S. Hwang, "A safety review on fuzzy-based relay selection in wireless sensor networks," *International Journal of Network Security*, vol. 17, no. 6, pp. 712–721, 2015.

[6] N. M. Haller, "On internet authentication," *Technical Report*, RFC 1704, Oct. 1994.

[7] N. M. Haller. "The s/key one-time password system," *Technical Report*, RFC 1760, Feb. 1995.

[8] M. S. Hwang, T. H. Sun, "Using smart card to achieve a single sign-on for multiple cloud services," *IETE Technical Review*, vol. 30, no. 5, pp. 410–416, 2013.

[9] M. Kumar, K. Dutta, I. Chopra, "Impact of wormhole attack on data aggregation in hieracrchical WSN," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 70–77, 2014.

[10] L. Lamport, "Password authentication with insecure communication," *Communication ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[11] 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.

[12] C. C. Lee, M. S. Hwang, and I-En Liao, "A new authentication protocol based on pointer forwarding for mobile communications," *Wireless Communications and Mobile Computing*, vol. 8, no. 5, pp. 661–672, 2008.

[13] C. C. Lee, I-En Liao, and M. S. Hwang, "An extended certificate-based authentication and security protocol for mobile networks," *Information Technology and Control*, vol. 38, no. 1, pp. 61–66, 2009.

[14] C. C. Lee, C. T. Li, and S. D. Chen, "Two attacks on a two-factor user authentication in wireless sensor networks," *Parallel Processing Letters*, vol. 21, no. 1, pp. 21–26, 2011.

[15] C. C. Lee, L. H. Li, and M. S. Hwang, "A remote user authentication scheme using hash functions," *ACM Operating Systems Review*, vol. 36, no. 4, pp. 23–29, 2002.

[16] 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.

[17] T. Maitra, R. Amin, D. Giri, P. D. Srivastava, "An efficient and robust user authentication scheme for hierarchical wireless sensor networks without tamper-proof smart card", *International Journal of Network Security*, vol. 18, no. 3, pp. 553–564, 2016.

[18] D. Manivannan, P. Neelamegam, "An efficient key management scheme in multi-tier and multi-cluster wireless sensor networks", *International Journal of Network Security*, vol. 17, no. 6, pp. 651–660, 2015.

[19] E. O. Osei, J. B. Hayfron-Acquah, "Cloud computing login authentication redesign," *International Journal of Electronics and Information Engineering*, vol. 1, no. 1, pp. 1–8, 2014.

[20] Y. B. Saied, A. Olivereau, "A Lightweight Threat Detection System for Industrial Wireless Sensor Networks", *International Journal of Network Security*, vol. 18, no. 5, pp. 842–854, 2016.

[21] H. Saini, "1-2 skip list approach for efficient security checks in wireless mesh networks," *International Journal of Electronics and Information Engineering*, vol. 1, no. 1, pp. 9–15, 2014.

[22] J. J. Shen, C. W. Lin, and M. S. Hwang, "A modified remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 414–416, 2003.

[23] G. Sharma, S. Bala, A. K. Verma, "An improved RSA-based certificateless signature scheme for wireless sensor networks," *International Journal of Network Security*, vol. 18, no. 1, pp. 82–89, 2016.

[24] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," *in Proceedings of ACM Workshop on Security of Ad hoc and Sensor Networks*, pp. 59–64, 2004.

[25] K. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 244–251, 2006.

[26] Q. Q. Xie, S. R. Jiang, L. M. Wang, C. C. Chang, "Composable secure roaming authentication protocol for cloud-assisted body sensor networks," *International Journal of Network Security*, vol. 18, no. 5, pp. 816–831, 2016.

Chung-Huei Ling received his M.S. in Applied computer science and technology from Azusa Pacific University, Azusa, California, USA in 1990 and M.B.A in accounting from Northrop University, Los Angeles, California, USA in 1989. He is currently pursuing the Ph.D. degree from Computer Science and Information Engineering Department of Asia University, Wufeng, Taiwan. His research interests include information security, cloud computing, and radio frequency identification.

Cheng-Chi Lee received the Ph.D. degree in Computer Science from National Chung Hsing University (NCHU), Taiwan, in 2007. He is currently an Associate Professor with the Department of Library and Information Science at Fu Jen Catholic University. Dr. Lee is currently as an editorial board member of International Journal of Network Security and Journal of Computer Science. He also served as a reviewer in many SCI-index journals, other journals, other conferences. His current research interests include data security, cryptography, network security, mobile communications and computing, wireless communications. Dr. Lee had published over 100+ articles on the above research fields in international journals.

Chou-Chen Yang received his B.S. in Industrial Education from the National Kaohsiung Normal University, in 1980, and his M.S. in Electronic Technology from the Pittsburg State University, in 1986, and his Ph.D. in Computer Science from the University of North Texas, in 1994. From 1994 to 2004, Dr. Yang was an associate professor in the Department of Computer Science and Information Engineering, Chaoyang University of Tech- nology. Currently, he is a professor in the Department of Management Information Systems, National Chung Hshing University. His research interests include network security, mobile computing, and distributed system.

Min-Shiang Hwang received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China, 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 project leader for research in computer security at TL in July 1990. He obtained the 1997, 1998, and 1999 Distinguished 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 database and data security, cryptography, image compression, and mobile communications.

Reliable Alert Fusion of Multiple Intrusion Detection Systems

Vrushank M. Shah[1], and A. K. Agarwal[2]
(Corresponding author: Vrushank M. Shah)
Department of Electronics and Communication, Indus University[1]
21/246 Parasnagar-2, Solaroad, Ahmedabad, Gujarat
Vice Chancellor, Gujarat Technological University[2]
(Email: vrushank26@yahoo.in)
(Received Jan. 27, 2016; revised and accepted Apr. 6 & Apr. 25, 2016)

Abstract

Alert Fusion is a process of combining alerts from multiple Intrusion Detection Systems to make a decision about the presence of attack or intrusion. A reliable decision from an alert fusion requires that Intrusion detectors involved in the fusion process generates fully reliable alerts. The unreliable alerts from intrusion detectors may completely misleads the decision making process. The existing alert fusion operators doesn't incorporate reliability of Intrusion detectors. In this work, we have proposed a novel alert fusion method which overcomes the limitations of existing fusion methods and fulfils the requirements for alert fusion domain. We have demostrated the results for two different approaches of deriving reliability value of intrusion system detector which are based on conflict and true positive rate of intrusion detectors. The results shows the robustness of proposed rule in fusing alerts from multiple intrusion detection system. Our proposed approach shows a drastic reduction in false positive rate without affecting the true positive rate.

Keywords: Alert fusion, DARPA99, IDS, KDD99, reliability

1 Introduction

Intrusion Detection system (IDS) is a security system that monitors the traffic on a computer network system, analyzes the traffic and generates a warning called as alert or alarm in case any abnormalities found [5, 11]. In this sense intrusion detection system (IDS) is defined as a classifier which collects the evidences about the presence or absence of an intrusion. The evidences collected are usually incomplete, uncertain, contradictory or conflicting and may be complementary. The use of single IDS as a detector has two major drawbacks: higher false alarm rate and lower intrusion detection coverage, these limits the detection performance of an IDS in presence of multiple categories of attack/intrusion.

An prospective approach of tackling with multiple categories of attack is through the use of distributed IDS [1]. The distributed IDS consists of multiple intrusion detection systems which are dissimilar in nature. They are dissimilar by the fact that they extract different features of network traffic or might have completely different detection algorithms, viz., signature based IDS or anomaly based IDS [3]. Unfortunately, Along with the potential benefits of distributed Intrusion detection system there are two major problems. The first problem is to decide an efficient fusion rule to combine the diverse evidences provided by this systems and second problem is to determine whether the evidence provided by these systems are reliable, i.e., finding reliability value of IDS involved in the fusion process. The reliability of intrusion detection system is defined as the amount of trust on the ability of IDS and the evidence provided by IDS. The value of reliability factor decides the discounting factor for discounting the evidences of conflicting, complementary and unreliable IDS. The classical method of fusing evidences from multiple intrusion detection systems assume all the IDS to be equally reliable and assign same weighage to each of the evidences. However, in real scenario it is not true because some IDS are dominant for detecting certain class of attack and also its evidence can be more reliable compared to other IDS involved in fusion process.

Our focus in these work is to overcome the limitations and issues in the method of fusing dissimilar evidence from multiple IDS and to derive the numerical value of reliability of IDS. In this paper, we propose a novel fusion operator that not only fuses the evidence but also incorporate the reliability value.

The rest of the paper is organized as follows: Section 2 introduces the traditional alert fusion flow and explains various fusion rules proposed in literature [2, 12, 17]. Section 3 discusses the requirements and limitations of an ideal fusion rule. Section 4 describes the proposed alert fusion approach and discusses various features of the pro-

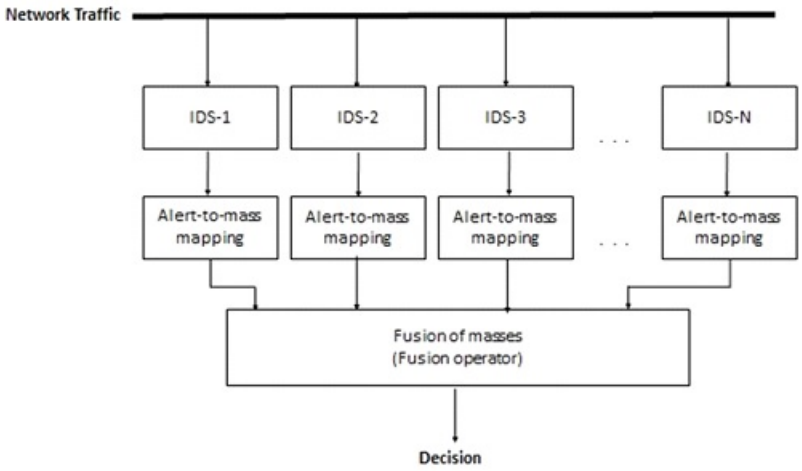


Figure 1: Traditional Alert Fusion Flow Diagram

posed alert fusion approach. Section 5 shows the simulation setup and describes the dataset. Section 6 shows results of proposed fusion rule under four different experiments. Finally in Section 7 we draw the conclusion.

2 Related Work

Figure 1 shows the traditional method for combining alerts from N different intrusion detection system. Each IDS sniffs the incoming network traffic and alerts for the presence of an attack. The alerts generated by IDS is converted to a mass value and all such masses are fused by fusion operator. This section will show the process of alert-to-mass mapping and gives a brief overview on existing fusion rule.

2.1 Alert to Mass Mapping

An IDS sniffs the traffic and generates positive and negative alerts. If we denote the hypothesis that attack is present by H and attack not present by -H then according to [6] we have,

$$\begin{aligned} m(H) &= \frac{P}{P+N+C} \\ m(-H) &= \frac{N}{P+N+C} \\ m(Hor-H) &= \frac{C}{P+N+C}, \end{aligned}$$

where, P- positive evidence in favor of hypothesis H, N-Negative evidence opposing the hypothesis H or favoring hypothesis -H and C is constant which is equal to 2 for binary frame of hypothesis. $m(H)$ is the mass value for hypothesis H. $m(Hor-H)$ is mass value for hypothesis H or $-H$ and can be called $m(uncertain)$ i.e, mass value for uncertainty between H and -H. Figure 2 shows the effect of increase in positive evidence on mass value of $m(H)$, $m(-H)$ and $m(uncertain)$.

2.2 Fusion Rules

The fusion rules are used to combine masses from n evidence sources and outputs a fused decision. For number of evidence sources $n \geq 2$ let $\Theta = \{\theta_1, \theta_2, \theta_3, \ldots, \theta_n\}$ be the frame of discernment for the fusion problem under consideration having n exclusive and exhaustive hypothesis . The sets of all subsets of Θ is called as power-set of Θ and is denoted by 2^{Θ}. The power-set is usually closed under unions, intersections and complements and is defined as a Boolean algebra. The fusion rules such Dempster shafer Rule in [12], Yager's Rule in [17] and Smet's TBM Rule in [13] are rules which are closed under Union operator. However, this rules doesn't contain intersections of element of Θ.

A basic belief assignment (BBA) is a function m from 2^{Θ}, the power set of Θ to [0,1]. The belief mass assignment will satisfy the property:

$$m(\phi) = 0 \text{ and } \sum_{A \in 2^{\Theta}} m(A) = 1.$$

Here, $m(\phi)$ is the mass assigned to null set. Let, $m_1(B)$ and $m_2(C)$ are two independent masses from two sources of evidence. Then the combined mass $m(A)$ is obtained by combining $m_1(B)$ and $m_2(B)$ through conjunctive rule,

$$\begin{aligned} m(A) &= \sum_{\substack{B,C \in 2^{\Theta} \\ B \cap C = A}} m_1(B) m_2(C) \\ m(\phi) &= \sum_{\substack{B,C \in 2^{\Theta} \\ B \cap C = \phi}} m_1(B) m_2(C). \end{aligned}$$

Disjunctive rule of combination is defined for union of elements of Θ. If $m_1(B)$ and $m_2(C)$ are two independent masses from two sources of evidence then the combined mass m(A) obtained by combining $m_1(B)$ and $m_2(C)$

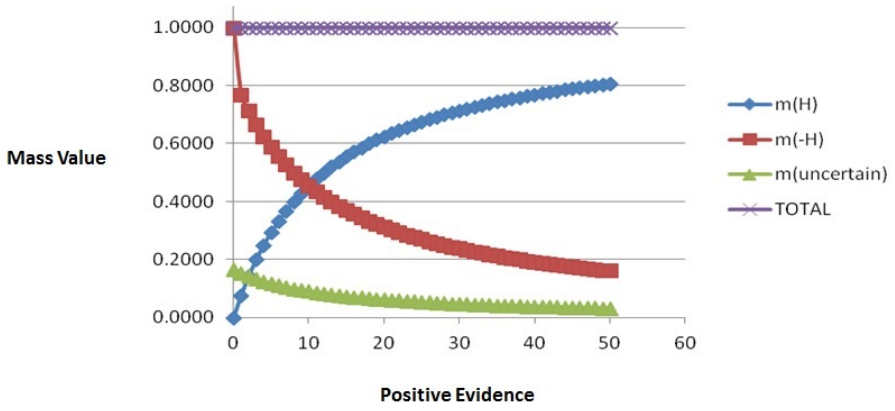


Figure 2: Effect of increase in positive evidence on mass value

through the rule,

$$m(A) = \sum_{\substack{B,C \in 2^{\Theta} \\ B \cup C = A}} m_1(B) m_2(C).$$

The disjunctive rule is preferable when some sources of evidence are unreliable but we don't know which one is unreliable.

The normalized version of conjunctive rule was proposed by Dempster and Shafer in [12] and is known as Dempster-Shafer rule. In DS rule, the fused masses m(A) is obtained from two independent sources of evidence $m_1(B)$ and $m_2(C)$ using following equation:

$$\begin{aligned} m(A) &= \frac{\sum_{\substack{B,C \in 2^{\Theta} \\ B \cap C = A}} m_1(B) m_2(C)}{1 - \sum_{\substack{B,C \in 2^{\Theta} \\ B \cap C = \phi}} m_1(B) m_2(C)} \\ m(\phi) &= 0. \end{aligned}$$

The above rule is defined for fusing two independent masses from sources of evidence. However, the same can be extended for n independent and equally reliable sources.

Dubois and Prade rule of combination by Dubois and Prade [2] is applicable when out of two sources, one source is unreliable and these unreliability is because of high conflict between the evidence they provide. DP rule assigns the value of conflict between two sources under union operator to the total mass value.

$$m(A) = \sum_{\substack{B,C \in 2^{\Theta} \\ B \cup C = A \\ B \cap C = \phi}} m_1(B) m_2(C) + \sum_{\substack{B,C \in 2^{\Theta} \\ B \cap C = A \\ B \cap C \neq \phi}} m_1(B) m_2(C).$$

3 Requirements and Limitations of Fusion Rules

Thomas in [15] suggests that the timely detection of intrusion in multiple IDS framework requires an efficient fusion rule that effectively combines evidence from multiple IDS and outputs a decision that accurately matches with existing ground truth. Following are the basic requirements for fusion rule as mapped out by authors:

- Fusion rule should incorporate the reliability of intrusion detection system for the evidence it provide about the presence of intrusion.
- The rule should be able to compromise between the reliable IDS and unreliable IDS.
- If all the IDS involved in fusion are unreliable then fusion rule should discard the available IDS and then new sets of IDS has to be found for concerned fusion problem.

According to Katar in [7] the quality of decision from a fusion operator varies application to application. In present work the goal is to combine alerts from multiple IDS systems, so the trustworthiness of alerts is a matter of concern. The existing fusion rules discussed in Section 2 have following limitations:

- None of the existing rule incorporates the reliability of source whose evidence are to be fused. Thus, there is no real time criteria which assign a numerical value of reliability to the evidence given by the source.
- The existing fusion rule considered all the sources of evidence to be equally reliable. However, in fusion framework there might be some unreliable sources which misleads to the fusion rule to give wrong decision.
- One major drawback related to the fusion rule as suggested by Goodman in [4] is that in an environment consisting of many hypotheses and many sources, it is difficult to decide whether to accept or reject the result of fusion rule. If sources of evidences are highly conflicting, the DS rule completely fails. If analyst blindly believes on the result then the decision can be misleading or complementary.

Table 1: DARPA 99 experiment description

Characteristic	Name
Dataset Name	DARPA 1999
Frame of Discernment (Θ)	[probe, -probe, θ]
Reliability criteria	TPR of IDS
No. of packets processed	5766

Table 2: Comparison of single IDS with fusion using DS and fusion using proposed rule by deriving reliability value using TPR

	Snort	Suricata	PHAD	NETAD	DS Rule	Proposed Rule
TP	127	124	144	118	131	143
TN	2715	2721	2730	2681	2644	5324
FP	2784	2778	2769	2818	2855	32
FN	140	143	123	149	136	267

The above requirements and limitations shows that we need a framework which can evaluate the numerical value of reliability of intrusion detection systems and discount the evidences based on their reliability beforehand. Also, there must be robust way as to handle conflict between sources and uncertainty assigned by sources to hypotheses.

4 Proposed Fusion Approach

To overcome the limitations and to match the requirements, we propose a novel method of fusing the evidences provided by source (Alerts generated by Intrusion Detection System). The flowchart of proposed fusion approach is as shown in Figure 3. The mass generated from alert to mass mapping block is used to derive reliability values along with CRF and DRF values. The input masses are discounted using this values and discounted masses are then fused using proposed rule. This section will explain the proposed rule, features of proposed rule and method for deriving reliability co-efficient of an Intrusion detection system.

4.1 Proposed Rule

The proposed rule is based on DS framework [12]. Here, $m_1(B)$ and $m_2(C)$ are two independent masses from two sources of evidence. Then the combined mass m(A) is obtained by combining $m_1(B)$and $m_2(C)$ through the rule,

$$m(A) = CRF(A) \sum_{\substack{B,C\in 2^{\Theta} \\ B\cap C=A}} m_1(B)m_2(C) + DRF(A) \sum_{\substack{B,C\in 2^{\Theta} \\ B\cup C=A}} m_1(B)m_2(C).$$

Where,

$$CRF(A) = \prod_n R_n$$

$$DRF(A) = (1 - \prod_n R_n)(1 - \prod_n (1 - R_n)).$$

Here, R_n is the reliability value of n^{th} source of evidence. $CRF(A)$ is conjunctive reliability value about A and $DRF(A)$ is disjunctive reliability value about A. CRF and DRF value acts as a weighting factor to compromise between conjunctive mass and disjunctive mass.

4.2 Features of Proposed Rule

The proposed rule effectively incorporates reliability of each source of evidence. If all the sources of evidence are reliable we get $CRF(A)$=1 and $DRF(A)$=0, so the proposed rule converge to conjunctive rule. If all the sources of evidence are unreliable we get $CRF(A)$=0 and $DRF(A)$=0, so the proposed rule does not give any solution and new sources of evidence has to be found. If some sources are reliable and some are unreliable we get $CRF(A)$=0.5 and $DRF(A)$=0.5, so the proposed rule will shows the compromise between conjunctive mass and disjunctive mass.

4.3 Deriving Reliability Co-efficient

One of the major problems of incorporating reliability of IDS into the fusion is problem of obtaining reliability coefficients. Reliability coefficients basically show a numerical value of trust in the mass value provided the Intrusion Detection system. The problem of finding reliability can be related to the problem of conflict between various Intrusion detection systems. The mere existence of conflict between the mass provided by Intrusion detection systems indicates the presence of an unreliable IDS which may cause the fusion result to be complementary from reality. An highly conflicting IDS will be assigned least reliability and least conflicting IDS will be assigned with highest reliability.

Another Approach of finding reliability is to relate reliability with the true alert rate of IDS. In these approach it is assumed that the IDS having highest true alert rate and lowest false alert rate will be assigned highest reliability and thereby, giving highest weightage in fusion process. While, all other IDS is assigned relative reliability value based on their true alert rate and false alert rate. The approach of assigning reliability based on true alert rate requires the ground truth knowledge. While, the approach of assigning reliability based on conflict between the IDS can work without knowledge of ground truth. In these work, we have use both the approaches and have compared result of proposed rule with existing rules.

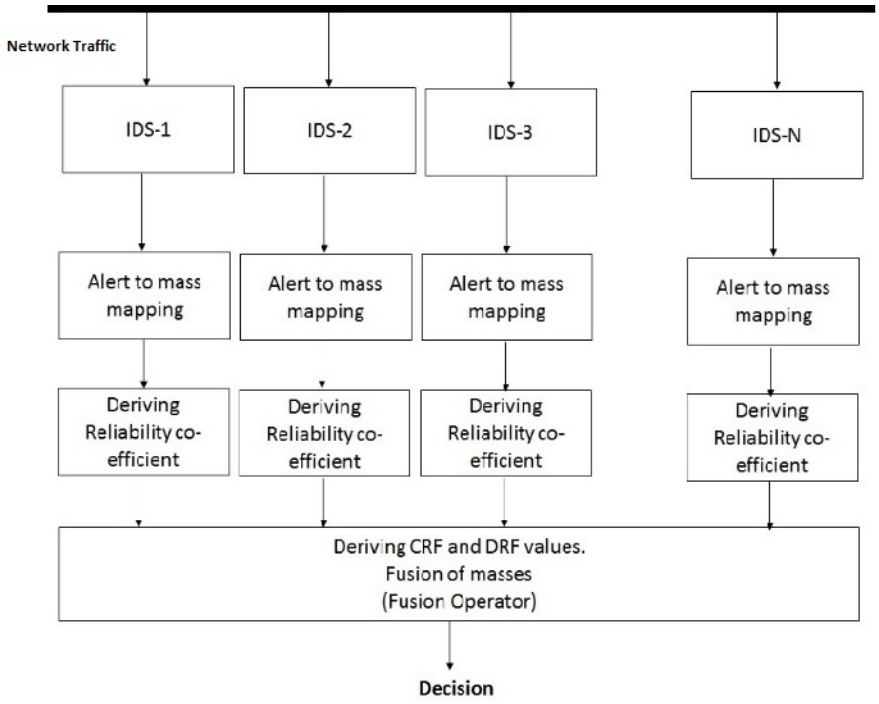


Figure 3: Flowchart of proposed fusion approach

Table 3: Comparison of single IDS with fusion using DS and fusion using proposed rule by deriving reliability value using TPR

	Snort	Suricata	PHAD	NETAD	DS Rule	Proposed Rule
TPR	0.4757	0.4644	0.5393	0.4419	0.4906	0.5356
FPR	0.5063	0.5052	0.5035	0.5125	0.5192	0.060
PPV	0.0437	0.0429	0.0494	0.0402	0.0439	0.8171
NPV	0.9510	0.9501	0.9569	0.9473	0.9511	0.9522
ACC	0.4929	0.4934	0.4984	0.4854	0.4813	0.9481

Table 4: DARPA 99 experiment description

Characteristic	Name
Dataset Name	DARPA 1999
Frame of Discernment (Θ)	[probe, -probe, θ]
Reliability criteria	Conflict between IDS
No. of packets processed	5766

Table 5: Comparison of single IDS with fusion using DS and fusion using proposed rule by deriving reliability value using conflict between evidences

	Snort	Suricata	PHAD	NETAD	DS Rule	Proposed Rule
TP	128	107	129	129	119	136
TN	2751	2718	2689	2768	2693	5130
FP	2748	2781	2810	2731	2806	244
FN	139	160	138	138	148	256

Table 6: Comparison of single IDS with fusion using DS and fusion using proposed rule by deriving reliability value using conflict between evidences

	Snort	Suricata	PHAD	NETAD	DS Rule	Proposed Rule
TPR	0.4794	0.4007	0.4831	0.4831	0.4457	0.3579
FPR	0.4997	0.5057	0.5110	0.4966	0.5103	0.0454
PPV	0.045	0.0377	0.0439	0.0439	0.0451	0.3579
NPV	0.9519	0.9444	0.9512	0.9525	0.9479	0.9525
ACC	0.4993	0.4899	0.4887	0.5024	0.4877	0.9133

5 Simulation Setup

5.1 Dataset Description

The MIT Lincoln Laboratory under the project DARPA has collected and distributed the first standard dataset for offline evaluation of IDS. DARPA98 and DARPA99 are two datasets available under DARPA project for the use of researchers. DARPA99 is modified and refined version of DARPA98 which consists of total 5 weeks of data which is divided into 3 weeks of training dataset and 2 weeks of testing dataset. Each week of dataset consists of five day of data from Monday to Friday of inside and outside traffic. The detailed explanation of various intrusions/attacks present in DARPA99 along with normal traffic is explained in detail in [8] by Kendall.

Most research in the field of IDS have been done using DARPA99 dataset. However, many of the researchers have criticized and argued about its applicability for IDS evaluation. Most of them consider that the dataset is very outdated and unable to create behavior like the present day attack. Along with the critics, there are significant argument in favor of DARPA99. In [15], Thomas argued that the non-availability of any other dataset that includes the complete network traffic was probably the initial reason to make use of the DARPA dataset for IDS evaluation by researchers. In [9], authors comment that if an present day advanced system could not perform well on DARPA dataset, it could also not perform acceptably on realistic data. Authors in [10] argued that even though there are shortcomings, the Lincoln evaluation indicates that even the best of the research IDS systems falls far short of the DARPA goals for detection and false-alarm performance. MCHugh in his work [10] believe that any sufficiently advanced IDS should be able to achieve good true positive detection performance on the DARPA IDS evaluation dataset. Demonstrating such performance, however, is only necessary to show the capabilities of such a detector, it is not sufficient.

The KDD99 dataset is knowledge discovery database originally created from DARPA98. The KDD99 dataset has 41 features along with one class label. The Class label consists of attack in four categories R2L, U2R, Probe and DOS. The complete details on types of attacks present in each categories and list of 41 features is available in the work by [14].

Authors in [14] suggested a new dataset called as NSL-KDD in order to solve the issues with KDD99. NSL-KDD was distributed for testing in year 2009 by University of New Brunswick. This new version of dataset does not have redundant records in train set, so the classifier does not get biased towards more frequent records. Also, the test set does not have duplicate records which give better detection rates. The reduced NSL-KDD make it reasonable to run the experiments without need of randomly selecting small set as in KDD99. The results by Tavallaee et al. [14] shows that the results obtained by NSL-KDD makes the evaluation results more consistent and comparable.

5.2 Selection of IDS

For alert fusion of multiple intrusion detection systems, we have selected four dissimilar IDS namely, Snort, Suricata, PHAD and NETAD. The reason behind such selection is that snort and suricata are signature based intrusion detectors while PHAD and NETAD are anomaly detectors. Thus, both categories are complementary to one another which enhances the performance of fusion system and within the category they are redundant which increases the accuracy.

5.3 System Configuration

The simulation environment consists three 3^{rd} Generation IntelCorei5processor (1.6GHz), Operating system installed is Linux Ubuntu with 4GB RAM. One machine deployed with Signature based IDS such as snort and suricata. Another Machine deployed with Anomaly detectors such as PHAD and NETAD. Third machine acts as an attacker machine having dataset loaded and is being replayed using TCPreplay.

6 Results

This section will discuss the results obtained under four different experiments namely, DARPA99 Experiment, KDD99 Experiment, NSL-KDD Experiment and some random experiments in order The performance metrics used to compare the results are true positive (TP), true negative (TN), false positive (FP), false negative (FN), true positive rate (TPR), false positive rate (FPR), positive prediction value (PPV), negative prediction value

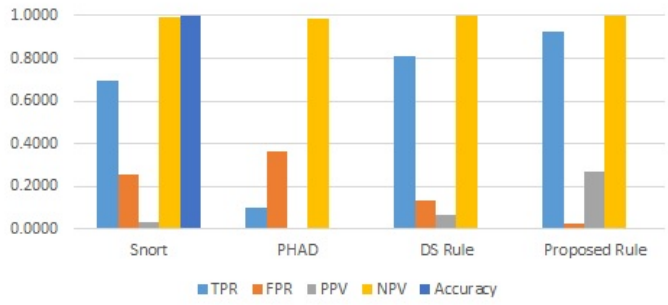


Figure 4: Comparison of proposed rule with DS rule against NSL-KDD for detecting R2L attack

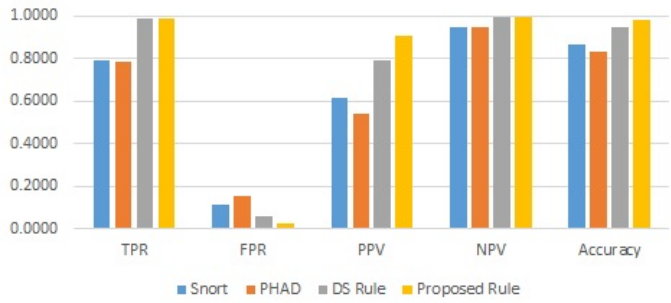


Figure 5: Comparison of proposed rule with DS rule against NSL-KDD for detecting DOS attack

(NPV) and Accuracy (ACC). The formal definition of each of this parameters are explained in Appendix A along with its significance.

6.1 DARPA99 Experiment

In DARPA99 Experiment, we preprocessed the dataset and total 5766 packets where loaded on to the network. In first experiment as per Table 1, we use the TPR of IDS as a reliability criteria. Table 2 and Table 3 shows the performance comparison of single IDS against the fusion using DS and fusion using proposed rule. The observed results shows an efficient reduction in number of false positives and an significant increase in the accuracy of IDS.

Table 4 shows the description of second experiment performed using DARPA99 where reliability is derived by calculating the amount of conflict between the IDS systems. Table 5 and Table 6 shows the comparison results of single IDS with fusion using DS and fusion using proposed rule by deriving reliability value using conflict between evidences.

Table 7: KDD 99 Experiment description

Characteristic	Name
Dataset Name	KDD 1999
Frame of Discernment (Θ)	[smurf, -smurf, θ]
Reliability criteria	TPR of IDS
No. of packets processed	3456

6.2 KDD99 Experiment

In KDD99 Experiment, we preprocessed the dataset and total 3456 packets containing attack and non-attack packet was loaded on the network and replayed using TCPReplay tool. The Frame of Discerment is selected to detect smurf attack. The total 1944 smurf attacks were present in processed dataset.

Figure 6: Confusion matrix of two IDS system having conflicting behavior

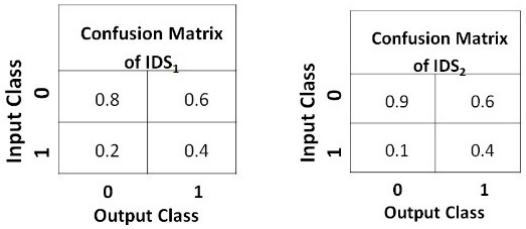


Figure 7: Confusion matrix of two IDS system having harmonious behavior

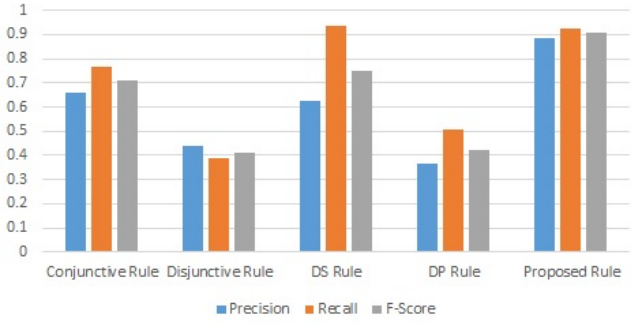


Figure 8: Comparing fusion rules under conflict behavior

Table 8: Comparison of single IDS with fusion using DS and fusion using proposed rule by deriving reliability value using TPR

	Snort	Suricata	PHAD	NETAD	DS Rule	Proposed Rule
TP	916	1015	982	910	969	1015
TN	788	763	769	762	765	1490
FP	724	749	743	750	747	22
FN	1028	928	962	1034	975	929

Table 9: Comparison of single IDS with fusion using DS and fusion using proposed rule by deriving reliability value using TPR

	Snort	Suricata	PHAD	NETAD	DS Rule	Proposed Rule
TPR	0.4712	0.5221	0.5051	0.4681	0.4985	0.5216
FPR	0.4788	0.4954	0.4914	0.4960	0.4940	0.0146
PPV	0.5545	0.5754	0.5693	0.5482	0.5647	0.9788
NPV	0.4339	0.4509	0.4443	0.4243	0.4243	0.4397
ACC	0.4931	0.5145	0.5067	0.4838	0.5017	0.7248

Table 10: KDD 99 experiment description

Characteristic	Name
Dataset Name	KDD 1999
Frame of Discernment (Θ)	[smurf, -smurf, θ]
Reliability criteria	Conflict between IDS
No. of packets processed	3456

Table 11: Comparison of single IDS with fusion using DS and fusion using proposed rule by deriving reliability value using conflict between IDS

	Snort	Suricata	PHAD	NETAD	DS Rule	Proposed Rule
TP	997	967	1015	960	1008	1033
TN	742	741	730	758	723	1501
FP	770	771	782	754	789	11
FN	947	977	929	984	936	911

Table 12: Comparison of single IDS with fusion using DS and fusion using proposed rule by deriving reliability value using conflict between IDS

	Snort	Suricata	PHAD	NETAD	DS Rule	Proposed Rule
TPR	0.5129	0.4974	0.5221	0.4938	0.5185	0.5314
FPR	0.5093	0.5099	0.5172	0.4987	0.5218	0.0073
PPV	0.5642	0.5564	0.5648	0.5601	0.5609	0.9895
NPV	0.4393	0.4313	0.4400	0.4351	0.4358	0.6223
ACC	0.5032	0.4942	0.5049	0.4971	0.5009	0.7332

Table 13: NSL-KDD experiment description

Characteristic	Name
Dataset Name	NSL-KDD
Frame of Discernment (Θ)	[R2L, -R2L, θ]
Reliability criteria	Conflict between IDS
No. of packets processed	5000
No. of R2L attacks present	52

Table 14: NSL-KDD experiment description

Characteristic	Name
Dataset Name	NSL-KDD
Frame of Discernment (Θ)	[DOS, -DOS, θ]
Reliability criteria	Conflict between IDS
No. of packets processed	5000
No. of R2L attacks present	944

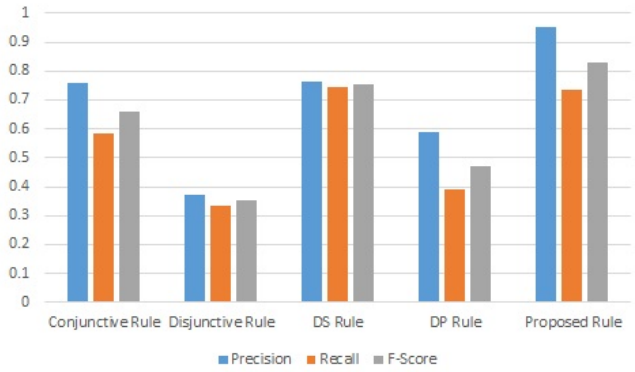


Figure 9: Comparing fusion rules under harmonious behavior

The KDD99 Experiment description is shown in Table 7. Table 8 and Table 9 shown the performance of proposed rule along with DS rule. Table 10 gives description about KDD99 experiment by using conflict between the IDS as a reliability criteria. Table 11 and Table 12 shows the results obtained under this experiment.

6.3 NSL-KDD Experiment

To perform evaluation of our proposed technique against NSL-KDD, We utilized two IDS systems out of four installed in the environment namely, snort and PHAD. In the work by Thomas [16], it is shown that PHAD has performs badly during detection of R2L and U2R attack. While, snort performs well against DOS and R2L categories of attack. In pre-processing of NSL-KDD using wireshark tool, it is found that DOS and R2L have very low variations and hence it is difficult to detect such attacks using traditional detection method.

Table 13 shows the description of NSL-KDD experiment for detecting R2L attack considering conflict as reliability criteria. Table 14 shows the description of NSL-KDD experiment for detecting DOS attack considering conflict as reliability criteria. Figure 4 shows the results of proposed rule with DS rule against NSL-KDD for detecting R2L attack and Figure 5 shows the results of proposed rule with DS rule against NSL-KDD for detecting DOS attack. It can be observed from the result that proposed rule gives highest accuracy and least false positive rate compared to individual IDS and DS rule.

6.4 Random Experiment

The behavior of proposed rule against existing rules is further checked with some random experiments. Here, we perform some random experiments under following situations:

- Conflicting Behavior;
- Harmonious Behavior.

We artificially generated the random packets but controlled them to have a conflicting behavior and harmonious behavior as shown in Figure 6 and Figure 7.The results for fusion of two intrusion system detectors having conflicting behavior is shown in 15. The proposed rule increases the accuracy by 20% compared to individual IDS. However, in the case of harmonious behavior the DS Rule improves accuracy by 10% while proposed rule improves it by 18% compared to individual IDS. Table 16 shows the results for harmonious behavior of IDS. Figure 8 and Figure 9 shows the results in terms of precision, recall and F-score under conflicting and harmonious behavior.

7 Conclusion

In this paper, a reliable alert fusion approach for combining alerts from multiple intrusion detection systems is proposed. The proposed rule incorporates reliability of intrusion detection during fusion process. The rule is designed to make compromise between conjunctive logic and disjunctive logic. The simulation was done against DARPA99, KDD99 and NSL-KDD and shows the performance of proposed approach with an improvement in false positive rate. We demonstrated the results for random situation under complementary and harmonious behavior to prove the robustness of our rule in terms of reducing false alert and enhancing accuracy of detection.

References

[1] G. G. Deverajan, R. Saravanan, "A novel trust based system to detect the intrusive behavior in MANET," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 31–43, 2015.

[2] D. Dubois and H. Prade, "On the combination of evidence in various mathematical frameworks," in *Reliability Data Collection and Analysis*, pp. 213–241, Springer, 1992.

Table 15: Comparison of proposed rule with existing rules under conflicting behavior

	IDS_1	IDS_2	Conjunctive Rule	Disjunctive Rule	DS Rule	DP Rule	Proposed rule
TP	1000	2500	1915	745	2341	820	2377
TN	2500	1000	1505	2141	1106	1960	2133
FN	1500	0	585	1160	159	800	190
FP	0	1500	995	954	1394	1420	300
TPR	0.4	1.0	0.766	0.3910	0.9364	0.5061	0.9259
FPR	0.0	0.60	0.398	0.3082	0.4424	0.5798	0.123
PPV	1.00	0.63	0.560	0.4384	0.6267	0.366	0.8879
NPV	0.625	1.00	0.720	0.648	0.874	0.710	0.9182
Accuracy	0.7	0.7	0.6847	0.5772	0.6894	0.556	0.9020

Table 16: Comparison of proposed rule with existing rules under harmonious behavior

	IDS_1	IDS_2	Conjunctive Rule	Disjunctive Rule	DS Rule	DP Rule	Proposed rule
TP	1000	1000	1461	800	1863	950	1906
TN	2000	2250	2036	1240	1929	1900	2303
FN	1500	1500	1039	1600	637	1490	691
FP	500	250	464	1360	571	660	100
TPR	0.40	0.40	0.58	0.33	0.75	0.39	0.73
FPR	0.20	0.10	0.19	0.52	0.23	0.26	0.04
PPV	0.67	0.80	0.76	0.37	0.77	0.59	0.95
NPV	0.57	0.60	0.66	0.44	0.75	0.56	0.77
Accuracy	0.60	0.65	0.70	0.41	0.76	0.57	0.84

[3] R. Goel, A. Sardana, and R. C. Joshi, "Parallel misuse and anomaly detection model," *International Journal of Network Security*, vol. 14, no. 4, pp. 211–222, 2012.

[4] I. R. Goodman, R. P. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*, vol. 37, Springer Science & Business Media, 1997.

[5] L. C. Huang and M. S. Hwang, "Study of intrusion detection systems environment," *Journal of Electronic Science and Technology*, vol. 4, p. 6, 2012.

[6] A. Jøsang, *Subjective Logic*, Book Draft, 2011. (`http://folk.uio.no/josang/papers/subjective_logic.pdf`)

[7] C. Katar, "Combining multiple techniques for intrusion detection," *International Journal of Computer Science and Network Security*, vol. 6, no. 2B, pp. 208–218, 2006.

[8] K. Kendall, "A database of computer attacks for the evaluation of intrusion detection systems," Technical Report DTIC Document, 1999.

[9] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection," in *Recent Advances in Intrusion Detection*, pp. 220–237, Springer, 2003.

[10] J. McHugh, A. Christie, and J. Allen, "Defending yourself: The role of intrusion detection systems," *IEEE Software*, vol. 17, no. 5, pp. 42, 2000.

[11] E. U. Opara, O. A. Soluade, "Straddling the next cyber frontier: The empirical analysis on network security, exploits, and vulnerabilities," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 10–18, 2015.

[12] G. Shafer et al., *A Mathematical Theory of Evidence*, vol. 1, Princeton University Press Princeton, 1976.

[13] P. Smets and R. Kennes, "The transferable belief model," *Artificial Intelligence*, vol. 66, no. 2, pp. 191–234, 1994.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications*, pp. 1–6, 2009.

[15] C. Thomas and N. Balakrishnan, "Improvement in intrusion detection with advances in sensor fusion," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 542–551, 2009.

[16] C. Thomas and N. Balakrishnan, "Performance enhancement of intrusion detection systems using advances in sensor fusion," *Supercomputer Education and Research Centre Indian Institute of Science*, Doctoral Thesis, 2009. (`http://www.serc.iisc.ernet.in/graduation-theses/CizaThomas-PhD-Thesis.pdf`)

[17] R. R. Yager, "On the dempster-shafer framework and new combination rules," *Information Sciences*, vol. 41, no. 2, pp. 93–137, 1987.

Vrushank Shah was born in Ahmedabad, Gujarat, India, in 1985. He received the B.E. degree in electronics and communication engineering from the North Gujarat

Appendix A

Table 17: Formal definition and significance of performance metrics

Name	Definition	Formula
True positive (TP)	Number of attacks that are correctly detected	-
False positive (FP)	Number of normal traffic packet that are incorrectly detected as attacks	-
True negative (TN)	Number of normal traffic packets that are correctly classfied	-
False negative (FN)	Number of attacks that are not detected	-
True positive rate (TPR)	Is the ratio of total true positives and sum of true positives with false negatives	$\frac{TP}{TP+FN}$
False positive rate (FPR)	Is the ratio of total false positives and sum of false positives with True negatives	$\frac{FP}{TN+FP}$
Positive prediction value (PPV)	Is the ratio of total true positives and sum of true positives with false positives.	$\frac{TP}{TP+FP}$
Negative prediction value (NPV)	Is the ratio of total true negative and sum of true negatives with false negatives.	$\frac{TN}{TN+FN}$
Accuracy (ACC)	Is the ratio of sum of TP and TN to the sum of TP, TN, FP and FN	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision(P)	Is a measure of what fraction of test data detected as attack is actually from the attack class	$\frac{TP}{TP+FP}$
Recall (R)	Is a measure of what fraction of attack class is correctly detected	$\frac{TP}{TP+FN}$
Fscore (F)	Is the balance between precision and recall	$\frac{2PR}{P+R}$

University, Patan, Gujarat in 2007 and the M.Tech. in communication systems engineering from Gujarat University, Ahmedabad in 2010. In 2010, he joined the Department of Electronics and Communication, L.J. Institute of Engineering and Technology as an Assistant Professor and in 2012, he joined Department of Electronics and Communication, Indus University, Ahmedabad as an assistant professor. Currently he is PhD Research Scholar at Gujarat Technological University and doing research in the field of Intrusion Detection system under the guidance of Dr. A. K. Aggarwal. His current research interests include Intrusion Detection system, Probability and Statistics, Decision making theory, Image processing and Speech Processing.

Honourable Dr. A. K. Aggarwal is currently the Vice Chancellor, Gujarat Technological University, Chandkheda, Ahmedabad. He received the B.Sc. degree in electrical engineering, Punjab University in 1964 and M.E. and PhD Degree in electrical engineering from M.S. University of Baroda in 1968 and 1981 respectively. He was with M.S. University of Baroda at different levels starting from Assistant professor, Associate professor, Professor and Head of electrical and computer science department from the 1967-1989. In year 1989, he joined as the Professor and Head of computer science department, Gujarat University. In 1999, he moved to Canada and joined University of Windsor as Associate professor. He has been honored with IEEE Millennium Medal and IEEE Outstanding Branch Counselor Award 1997. In 1999, he was the chair, IEEE India Council. His research interests involves cyber security and computer networking.

A Secure Communication Model for Expressive Access Control Using CP-ABE

Jayam Modi, Manav Prajapati, Abhinav Sharma, Ravi Ojha, and Devesh Jinwala
(Corresponding author: Devesh Jinwala)

Computer Engineering Department, S V National Institute of Technology, India
Computer Engineering Department, S V National Institute of Technology, India
Ichchhanath, SURAT-395 007, Gujarat, India
(Email: dcjinwala@gmail.com)
(Received June 13, 2015; revised and accepted Jan. 11 & Mar. 3, 2016)

Abstract

Attribute Based Encryption is a technique that associates user's attributes with keys. Data is encrypted using a specific policy and only those keys whose attributes satisfy that policy are allowed to decrypt it. In this paper, we propose a secure communication model based on Ciphertext Policy Attribute Based Encryption (CP-ABE). This model allows Role Based Access Control for documents without the use of a secure server to enforce the access policies. We propose a scalable implementation for key revocation and user attribute updation with improved flexibility. Our method uses a key revoke-list and key-version to achieve this. We show the implementation using the CP-ABE toolkit, an open source library that implements the CP-ABE scheme. We also show how confidentiality, integrity and source authentication is achieved in our model.

Keywords: Access rights, CP-ABE, expressive access control, secure communication model

1 Introduction

Information has been a valuable resource ever since humans began to communicate and like all other resources it needs to be protected. With the advent of the Internet and computing technology, digital means for exchanging information gained importance. Millions of people connected to Internet exchange information of potentially crucial nature. The methods used to secure this transfer of information have evolved over the years.

In this era of Internet, it is inevitable for various service providers like Google and Facebook to store sensitive personal information of users on servers. Considering the variety and importance of this information, there is a risk of an attack on these servers. This leads to concerns about compromise of personal data. To avoid such a compromise, the data can be stored in an encrypted form on the servers. This ensures that the privacy of the data remains intact. The task of selective sharing of information and access control now becomes a big challenge. Traditionally, a trusted server used to be employed in order to enforce access control but the data must be stored in unencrypted form on such a server. Public Key Infrastructure can be used to enforce access control over encrypted data by creating a trust model as discussed in [10]. In a PKI based model, when a user wants to selectively share the data, he must encrypt it with the public keys of each and every intended recipient. This is not a feasible option in many scenarios. When data is to be shared with a large group of users or the intended target audience determined by some attributes is not fully known, the PKI based model cannot be used. Another problem with this model is that the users higher in the access hierarchy have to store a large number of keys. This problem was solved to some extent using the method proposed in [1].

In [17], the idea of Attribute Based Encryption was introduced. Several schemes were proposed to achieve fine grained access control [9]. These scheme overcame the limitations faced by the model proposed in [10]. In an ABE based model, the data can be stored in an encrypted form on a server. This breakthrough lead to further developments in Role Based Access Control(RBAC) using ABE. Consequently, CP-ABE and KP-ABE encryption schemes were developed. A survey of these ABE schemes and access structures and their comparisons in cloud environment has been given in [12]. When the attributes are at different levels, the CP-HABE, which is a hierarchical CP-ABE scheme proposed in [14] proves to be useful.

1.1 Our Contribution

In this paper, we propose a secure communication model that can be used for selective sharing in an unsecure storage server environment. We achieve this by using CP-

ABE. Our model provides a scalable method for revoking keys and updating attributes of keys. We use a similar key revocation technique as in [6]. We attempt to address some issues in key revocation and dynamic attribute updation. Our scalable key revocation mechanism allows to effectively revoke a user's key immediately without much overhead by using revoke list. But the revocation process is completed lazily for a large batch of revoked keys. Our proposed mechanism provides complete freedom of choosing when to perform this lazy completion. It can be done when a threshold number of revocations have accumulated or any arbitrarily set time period has elapsed. We enable key attribute updation using the revocation mechanism itself. In our scheme, all key attributes can be assumed to be dynamic in nature. Also, our scheme does not require changing the public and master key pair of CP-ABE. We discuss the working of the proposed model in Section 3. We also provide pseudocodes in Section 4 to show clearly how our model can be implemented. We show how our model ensures confidentiality, integrity and authentication under some attack scenarios in Section 5.

2 Theoretical Background and Related Work

2.1 Theoretical Background

Cryptography is used to secure the communication between two parties. The earliest form of cryptography was secret key cryptography, which involved the use of a secret key that was known to both the parties before the exchange of data. As the size of networks and organizations grew bigger, the quadratic growth in the number of keys required for secure communication lead to serious concerns. Thus, when the idea of public key encryption was proposed by Diffie and Helman in [7], it was promptly accepted and as a result many different public key encryption schemes were developed.

A few emerging applications like cloud services often demand that the access to data should be governed by a policy wherein only specific individuals are granted access to the data. In such cases, there is a need for a cryptographic scheme that allows only those users whose attributes satisfies a decryption policy to decrypt the data. In Public Key cryptography, there is a single Private Key that can decrypt the data encrypted by the corresponding Public Key. Attribute Based Encryption [17] was introduced as an attempt to overcome this limitation.

Shamir in [18] defined a technique known as Identity Based Encryption that enabled any pair of users to communicate securely and to verify each other's signatures without exchanging private or public keys, without keeping directories and without using the services of a third party. In this scheme, the public key of the receiver is a combination of the receiver's attributes and it is computed by the sender with the help of publicly known attributes of the receiver. This eliminated the need for key exchange and therefore prevented man-in-the-middle attacks as opposed to public-key schemes. The private key of a user is generated by the key generation center after proper identity check of the user.

Attribute-based encryption (ABE) was an approach proposed by Sahai and Waters in [17]. In traditional public-key cryptography, a message is encrypted for a specific receiver using the receiver's public-key. But in large organizations, often there is requirement of a technique that allows members to specify access policies for restricting data to groups of intended recipients. This can be achieved by using a trusted server to store data. The server can check certification of a user before granting him access to files. A major drawback of this method is the security of the server. ABE aims to achieve secure selective sharing while removing the dependency on servers with access control mechanisms. The access control logic is embedded in the encryption technique and thus encrypted data remains confidential even if the storage server is untrusted.

There are two types of Attribute Based Encryption, namely Key-Policy Attribute Based Encryption (KP-ABE) and Ciphertext-Policy Attribute Based Encryption (CP-ABE). [9] provides a scheme to implement KP-ABE. In KP-ABE, ciphertexts are associated with sets of descriptive attributes, and users' keys are associated with policies. In key-policy ABE, the encryptor exerts no control over who has access to the data it encrypts, except by it's choice of descriptive attributes for the data. Rather, it must trust that the key-issuer issues the appropriate keys to grant or deny access to the appropriate users.

CP-ABE was first presented in [3]. In CP-ABE, a user's private key is associated with a set of attributes and the access policy is specified in the ciphertext. A user can decrypt an encrypted text if and only if his attributes satisfy the policy specified in the ciphertext. The policy can be built using conjunctions, disjunctions and (k, n) threshold gates. The private keys can be obtained by a user even after the data has been encrypted. Thus the actual set of users that can decrypt a ciphertext is not needed to be known at the time of encryption. This allows the incorporation of future users who may obtain a key that will satisfy the policy of the encrypted text and hence be able to read the data.

All the above efforts are shown to be special cases of Functional Encryption [4]. The term Functional Encryption was first seen in [11], disguised in the form of predicate encryption. Functional encryption is a scheme which allows a user to gain knowledge about a specific function of the encrypted text. The data is encrypted with a public key pk . A master secret key is held by a trusted authority. It can generate secret key sk_f corresponding to function f. The user having sk_f can compute the func-

tion f on any encryption of x. There are four phases in a functional encryption system - setup, keygen, encryption and decryption. The setup phase generates a public key and the master secret key. Keygen phase generates the secret key sk_k. Encryption phase encrypts a message x with the public key. Decryption phase enables user to compute $F(k, x)$ from the encrypted message. Functional encryption systems have a wide range of applications today like spam filtering on encrypted mail, expressive access control and mining on large datasets.

2.2 Related Work

The issue of user key revocation in CP-ABE is not addressed often in the proposed schemes. Majority of the authors who propose schemes focus on proving the security of their scheme. Piretti, Traynor and McDaniel in [15] roughly addressed the issue of attribute revocation for the first time. They suggested that each attribute should be valid only within a particular time-frame. After the validity of the attribute expires, the system administrator will release latest version of the attribute. The user updates his key based on the latest available version of the attribute. To revoke an attribute, the latest version of the attribute will not be released. The major problem with this solution is that of time synchronization between the system administrator and the user. To overcome this shortcoming, Bethencourt in [3] proposed that every key of a user should have an expiration date. A user will be able to decrypt the message only if the date of encryption of message is less than the expiration date of the user's key.

In [20], it is proposed that whenever an attribute needs to be revoked, the key generation authority will redefine the master key components of the revoked attributes. The corresponding public key components are also redefined. The user's secret keys need to be updated for data access. The new data is encrypted using the new public key. To perform these updates, proxy re-key's [13] are generated by the authority. Using these re-key's, the proxy servers can update the existing ciphertexts on the storage sever as well as the user's secret keys. This maintains backward compatibility in the system. This method transfers the load of the authorities onto the proxy servers leading to better performance than the methods proposed in [3] and [15].

Chen and Gerla in [5] proposed a fading function based method for implementing dynamic attributes. In their method, the concept of a fading function, F(x,y) was introduced. This function takes two parameters as input, the attribute name and the time at which its value is to be determined. It then outputs a unique value based on these two parameters. If the sender sends a message at time $t1$, the receiver will be able to decrypt that message at time $t2$ if and only if F(attribute, $t1$) = F(attribute, $t2$).

Weber in [19] proposed a method for incorporating the type of dynamic attributes whose values can be expressed in a list. He proposed that each attribute should be converted into a group element and then those elements should be translated into appropriate components of the private key. These components are transferred to the users device and stored in a secured compartment of the device that cannot be accessed by the user. All these attributes are bound together using a common random factor during the key generation process resulting in blinding of each key component. This in turn also blinds the ciphertext when the attributes are used in it. During the decryption process, the same common random factor is used to unblind the ciphertext. The malicious users cannot combine the components of the different keys in the same manner as an authentic user and thus the decryption algorithm will fail.

Chuha, Roy and Stoev in [6] use the concept of negative attributes to allow immediate key revocation, and the revocation process is completed lazily after a fixed time slice expires. To handle key attribute update, they propose that two separate access tress should be used for encryption process. One tree is for static attributes and other is for dynamic attributes. The tree of dynamic attributes is connected via a dummy node in the main access tree. After encryption using the main tree, the part of ciphertext that corresponds to dynamic attributes is separated out and re-encrypted using the access tree for dynamic attributes. The receiver applies original key to remaining part of ciphertext and obtains a key from the local key server for decrypting the dynamic attributes part of the cipher-text. Our model uses a similar technique but allows more flexibility and efficiency for key revocation by providing a mechanism to complete the revocation process after an arbitrary time limit or upon reaching a threshold number of revocations. Also, our scheme uses a simple technique for key attribute updation using the revocation mechanism. This allows making all key attributes dynamic.

Doshi in [8] proposed that for updating an attribute, the user should return his old secret key and the CA will give the user the new secret key to the user after verifying the new value of the attribute. The keygen algorithm takes old values from old secret key. An algorithm for using this technique in semi-trusted environment is also discussed in it.

3 Proposed Secure Communication Model

We propose a Secure Communication Model that allows expressive access control without the use of a secure storage server. The model has been developed using the cpabe-toolkit [2].

The CP-ABE toolkit provides four command line tools to perform the various operations of the CP-ABE scheme proposed in [3]. They can be used manually or can be invoked by larger systems. The four command line tools are:

- cpabe-setup - generates a public key and a master secret key;
- cpabe-keygen - generates a private key with a given set of attributes;
- cpabe-enc - encrypts a file according to a policy, which is an expression in terms of attributes;
- cpabe-dec - decrypts a file using a private key.

Our proposed model provides the following functionalities:

- Sending file to intended audience - A user can specify the attributes of the intended audience while sending a file. The model ensures that only the intended audience will be able to view the file.
- Receive files - A user can receive the files intended for him.
- Revoke access rights of some user - This is necessary if some user is no longer a part of the network and should not have access to the network's files.
- Update access rights of some user - This is necessary if the role of some user in the network changes.

Figure 1 shows a use case diagram of the model.

3.1 Components of the Proposed Model

The model has three entities, namely the Repository, Key Generation Center (KGC) and the users. The Repository and the KGC interact with the users to perform several tasks.

The Repository is a central server accessible to all. It is assumed that this server is not secure. The repository stores the encrypted files sent by all users, along with the timestamp when each file was uploaded at server, sending user's id and the minimum `Key_Version` required to decrypt the file. The Repository has a Public-Private key pair. All data sent from repository is signed using its Public key.

The Key Generation Center (KGC) performs the tasks related to key management. It stores the user's attributes and performs tasks such as initial key distribution, key revocation, distributing `Revoke_List` and `Active_Key_Version`, key renewal and updating user's attributes. The KGC has a Public-Private key pair. All data sent from KGC is signed using its Public key.

Each user has access to KGC and Repository. They have their own private key (i.e. cp-abe secret key), KGC and Repository's public key and the CP-ABE public key with them.

A private key in CP-ABE is associated with a set of attributes. In our proposed model, each private key has two types of attributes - User Attributes and Essential Attributes. User Attributes describe the user. E.g. department, name, experience, salary, etc. Essential Attributes are used to implement the model features. They are key_id and `Key_Version`. For a key to decrypt a file, in addition to satisfying the constraints on user attributes, it also needs to satisfy the constraints on essential attributes.

Each user has a `user_id` which uniquely identifies him within the organization. Each key has a `key_id` that uniquely identifies a key. At any given point of time, each `user_id` may be associated with only one `key_id`. Each `key_id` is uniquely associated with a fixed set of attributes. So, the `key_id` associated with a user has to be changed if the attributes of the user are changed.

At any point of time, the whole system will have an `Active_Key_Version`. It is a positive integer that is used for implementing the key revocation feature of the model. It starts from 1 and can only be incremented. Every key also has a `Key_Version` as one of its attributes which may be less than or equal to the `Active_Key_Version`.

3.2 The Proposed Model

When a new user enters the organization, he is authenticated at the KGC. His attributes are stored in a database at the KGC and a private key is provided to him. The public key of KGC and Repository is also provided to him. It is assumed that these functions are done through direct physical contact. A user must maintain the secrecy of his private key.

Further updates in the private key don't require direct physical contact as the KGC can simply encrypt the updated private key using cp-abe with policy such that only the concerned user may be able to decrypt it.

3.2.1 Send a File

For encrypting a file, the following data is required:

1) cp-abe Public key - It is publicly available and each user has a local copy on their own machine.
2) `Revoke_List` - This is the list of `key_ids` which have been revoked since the last `Active_Key_Version` update.
3) `Active_Key_Version` - This is the `Active_Key_Version` as discussed in Section 3.1.

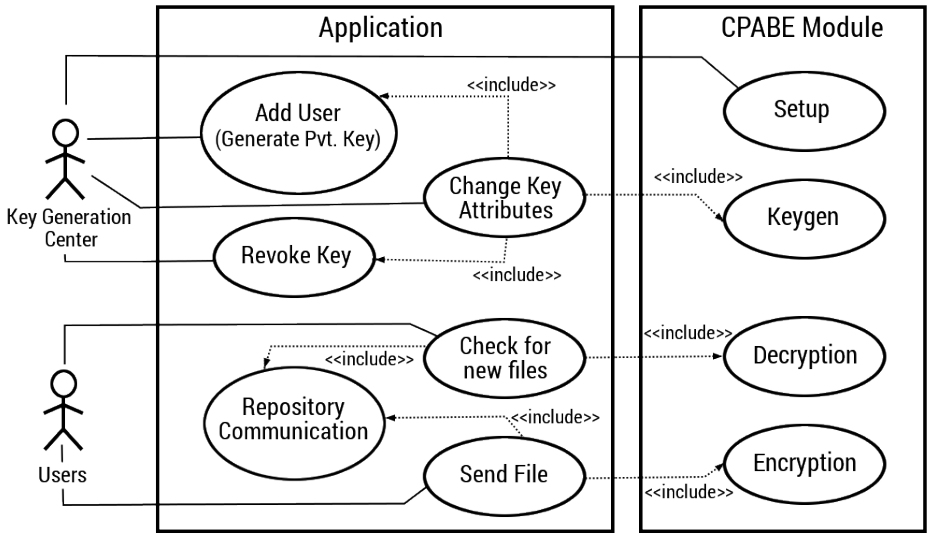


Figure 1 Use case diagram for the communication model

The `Revoke_List` and `Active_Key_Version` are stored at the KGC and are fetched every time a file needs to be encrypted.

The following steps are followed when a user uploads a file to repository:

1) User obtains the current `Revoke_List`, and `Active_Key_Version` from the KGC.
2) User select the policy file. The policy file contains a boolean formula that describes the User Attributes of the intended audience.
3) User generates the augmented policy file. In this step, the boolean formula in the policy file is augmented with constraints on the essential attributes. It includes the following:
   - minimum `Key_Version` required to decrypt the file, which is the `Active_Key_Version`. E.g. If the `Active_Key_Version` is 2, then version >= 2 is used;
   - list of revoked `key_ids`. The `key_id` that tries to decrypt the file should not be any of these. E.g. If the `Revoke_List` is (4, 6, 10), then (`key_id` != 4 and `key_id` != 6 and `key_id` != 10) is used.

   Thus, the (augmented policy) = (original policy) and (constraints on essential attributes). E.g. (augmented policy) = (original policy) and ((`key_id` != 4 and `key_id` != 6 and `key_id` != 10) and version >= 2).
4) User encrypts the selected file using cp-abe with policy as the augmented policy generated.
5) User sends 'Upload Request' to Repository along with his `user_id`.
6) Repository generates Asymmetric Key Pair (K1, K2) such that K1 and K2 are inverse of each other as done in RSA [16].
7) Repository encrypts K1 using cp-abe with policy such that only the concerned user's key be able to decrypt it.
8) Repository signs the encrypted data with his Public Key and sends it to the user.
9) User receives signed and encrypted K1 from Repository. He verifies the signature and decrypts K1.
10) User signs the encrypted file that he wants to upload with K1.
11) User uploads the signed and encrypted file along with the minimum `Key_Version` required to decrypt (which is the `Active_Key_Version` at the time of encryption) to the repository.
12) Repository receives the file, verifies the signature using K2 and stores it along with the upload timestamp and the minimum `Key_Version` required to decrypt the file.

It should be noted that a key with `Key_Version` less than the `Active_Key_Version` at the time of encryption of a file will not be able to decrypt the file. Such a key should be renewed before trying to use it to decrypt a file.

3.2.2 File Refresh

The user may periodically check the repository for new files intended for him. The following steps are performed in this operation:

1) User sends the File Refresh Request to the repository. In the request, the user also sends the timestamp of the last file refresh done by him.
2) Repository selects all files that were uploaded after the given timestamp along with the upload timestamp and the minimum `Key_Version` requirement of each file. Repository signs this data with his Public Key and sends the signed data to user.
3) User receives data from Repository and verifies the signature.
4) If the maximum of `Key_Version` requirement of all received files is greater than the `Key_Version` of the key possessed by the user, then he requests for an updated key from the KGC otherwise the next step is skipped.
5) (skipped if not required) The KGC generates another private key using cp-abe module for the user using his attributes (that were stored at KGC) and the `Active_Key_Version` of system. KGC encrypts the new key using cp-abe with policy such that only the concerned user be able to decrypt it. KGC signs the encrypted key with its Public Key and sends it to user. If the `key_id` of the user has been revoked, the KGC doesn't return any new key.
6) (skipped if `key_id` was revoked) User receives the signed and encrypted updated key from KGC. He verifies the signature and decrypts it with his old key.
7) User tries to decrypt each file using cp-abe module one by one using his private key (which may or may not be updated in the above step).
8) User deletes the files that couldn't be decrypted and can view those that were successfully decrypted.

3.2.3 Key Revocation

The KGC receives request from the administration for revoking a certain key. The following steps are performed at the KGC:

- KGC adds the `key_id` to the `Revoke_List`. Now whenever, a user requests the `Revoke_List`, this new list will be sent. So, when the user encrypts the file, the augmented policy will make sure that the none of the revoked keys can decrypt the file.
- It marks the `key_id` in its database.

This is a temporary fix for revoking keys as it is not scalable. The size of the `Revoke_List` will keep on increasing and lead to increased overhead. When a certain number of revoke keys have accumulated OR a fixed time period has passed, the following process will be done by KGC:

1) KGC increments the `Active_Key_Version` of the system.
2) KGC sets the `Revoke_List` to empty.

Now, whenever a file is encrypted, the new `Active_Key_Version` will be used to construct the augmented policy. As the existing keys have old `Key_Version`, they will not be able to decrypt it. The users may then ask the KGC for key renewal. They will be given their new keys, which will have the same attributes as their old key, but with the `Key_Version` incremented. The revoked users marked in the database will not be issued new keys.

Note that the user can still use his revoked key to decrypt only those messages that had been encrypted before his key was revoked.

3.2.4 Update Attributes

When the KGC receives request to update attributes of a particular user, the following steps are performed:

1) KGC finds the current `key_id` associated with the concerned user and revokes that key.
2) KGC generates a key with a new `key_id`, new attributes and `Active_Key_Version`.
3) KGC encrypts the updated key using cp-abe with policy such that only the old key of concerned user be able to decrypt it.
4) KGC signs the encrypted updated key. KGC sends the data to user.
5) User receives the data, verifies the signature and decrypts the updated key using his old key.

Note that even after receiving his new updated key, the user still possesses his old key. This old key can be used to decrypt only those messages that had been encrypted before his attributes were updated.

4 Pseudocodes

This model has been implemented using socket programming. There are three modules - User, Repository and KGC. The KGC and Repository modules run on a server and service requests sent by User module. The user can invoke commands for sending files or doing a file refresh through his modules. Calls are made to the CP-ABE toolkit to perform various functions.

The list of functions, invoked in various pseudocodes, along with their description is as follows:

- **cpabe-keygen(masterkey, public_key, attributes)** - CP-ABE module function that returns a private key associated with given attributes.

- **cpabe-enc(public_key, plain_text, policy)** - CP-ABE module function that returns encrypted file with given policy.
- **cpabe-dec(public_key, private_key, encrypted_text)** - CP-ABE module function that returns decrypted file if the provided key satisfies the policy.
- **request(request_type, receiver, ...)** - sends the specified request to the receiver with optional arguments and returns response from receiver.
- **response(data)** - sends data in response to the current request.
- **send(data, receiver)** - sends the data to specified receiver.
- **receive(data)** - receives data from current connection.
- **verify(attributes)** - verify if the passed attributes are correct.
- **new_keyID()** - generates a new unique key_id.

The following notations are obeyed in the pseudocodes:

- $\mathbf{(msg)_{k1}}$ - msg encrypted using Public Key cryptography with key k1. Denotes encryption, if k1 is public key. Denotes signing, if k1 is private key.
- $\mathbf{\{msg\}_{pol}}$ - msg encrypting using cpabe-enc with policy pol.

4.1 Repository Module

1) **Response to File Refresh Request** - This procedure in Pseudocode 1 is invoked while receiving new files.

Pseudocode 1 Response to Refresh Request

Input:
last_refresh_TS : Timestamp of last file refresh done by user

procedure PROCESS_REQUEST('Refresh Messages', last_refresh_TS)
  pkt ← ' '
  for all msg whose TS is > last_refresh_TS **do**
    pkt ← pkt + (msg, sender_user_id, upload_TS, key_version_required)
  end for
  signed_pkt ← $(\text{pkt})_{\text{REPO\_priv\_key}}$
  Response (signed_pkt)
end procedure

2) **Response to Upload Request** - This procedure in Pseudocode 2 is invoked when the user sends File Upload Request.

Pseudocode 2 Response to Upload Request

Input:
user_id : user_id of the user who sent Upload Request

procedure PROCESS_REQUEST('Upload Request', user_id)
  (K1, K2) ← generate Asymmetric key pair
  sender_key_id ← key_id_map[user_id]
  policy ← 'key_id=sender_key_id'
  $\{\text{K1}\}_{\text{policy}}$ ← cpabe-enc (pub_key, K1, policy)
  signed_msg1 ← $(\{\text{K1}\}_{\text{policy}})_{\text{REPO\_priv\_key}}$
  Response (signed_msg1)
  Receive (signed_msg2)
  $\{\text{msg}\}_{\text{policy}}$ ← $(\text{signed\_msg2})_{\text{K2}}$
  store ($\{\text{msg}\}_{\text{policy}}$, Current TS, Key_version, user_id)
end procedure

4.2 User Module

1) **Essential Constraints Generation** - This procedure in Pseudocode 3 is invoked by send file method.

Pseudocode 3 Essential constraints generation

Input:
Revoke_List : List of key_id that are revoked
AKV : Active_Key_Version of the system

procedure ESSENTIAL_CONSTRAINTS_GEN (Revoke_List, AKV):
  essential_constraints ← ' '
  for all x in Revoke_List: **do**
    essential_constraints ← essential_constraints + 'and key_id ! =x'
  end for
  essential_constraints ← essential_constraints + 'and key_version >= AKV'
  return essential_constraints
end procedure

2) **Send File** - This procedure in Pseudocode 4 is invoked when the user decides to upload file to Repository.

3) **File Refresh** - This procedure in Pseudocode 5 is invoked when the user wants to receive new files.

Pseudocode 4 Send file

Input:

msg : File that is to be upload to Repository

policy : Boolean formula denoting which users the file is intended for

1: **procedure** SEND_FILE(msg, policy):
2: signed_pkt1 ← Request ('Revoke_List and Active_Key_Version' , KGC)
3: (Revoke_List, AKV) ← $(\text{signed\_pkt1})_{\text{KGC\_pub\_key}}$
4: essential_constraints ← Essential_constraints_gen (Revoke_List, AKV)
5: augmented_policy ← '(' + policy + ')' + essential_constraints
6: $\{\text{msg}\}_{\text{augmented\_policy}}$ ← cpabe-enc (pub_key, msg, augmented_policy)
7: signed_pkt2 ← Request ('Upload Request', REPO, user_id)
8: $\{\text{K1}\}_{\texttt{key\_id=sender\_id}}$ ← $(\text{signed-pkt2})_{\text{REPO\_pub\_key}}$
9: K1 ← cpabe-dec (pub_key, priv_key, $\{\text{K1}\}_{\texttt{key\_id=sender\_id}}$)
10: signed_msg ← $(\{\text{msg}\}_{\text{augmented\_policy}})_{\text{K1}}$
11: send (signed_msg, REPO)
12: **end procedure**

4.3 KGC Module

1) **Response to Revoke_List and Active_Key_Version request** - This procedure in Pseudocode 6 is invoked when the user sends Request for `Revoke_List` and `Active_Key_Version`.

2) **Response to Attribute Updation** - This procedure in Pseudocode 7 is invoked when KGC has to update attributes of the user.

3) **Response to Update `key_version` Request** - This procedure in Pseudocode 8 is invoked when the user sends Update `key_version` request.

4) **Response to revoke user key request** - This procedure in Pseudocode 9 is invoked when a user's key is to be revoked.

Pseudocode 5 File Refresh

Input:

CKV : Current_Key_Version of key possessed by the invoking user.

last_refresh_TS : Timestamp of last File Refresh done by the user.

1: **procedure** REFRESH(CKV, last_refresh_TS):
2: signed_pkt ← Request ('Refresh Messages', REPO, last_refresh_TS)
3: Message_list ← $(\text{signed\_pkt})_{\text{REPO\_pub\_key}}$
4: Required_version← Max (version requirement of all files)
5: **if** Required_version > CKV **then**
6: signed_pkt ← Request ('Update Key_Version', KGC, user_id)
7: $\{\text{new\_key}\}_{\texttt{key\_id = requester\_key\_id}}$ ← $(\text{signed\_pkt})_{\text{KGC\_pub\_key}}$
8: new_key ← cpabe-dec (pub_key, priv_key, $\{\text{new\_key}\}_{\texttt{key\_id=requester\_key\_id}}$)
9: **end if**
10: **for all** enc_msg in Message_list **do**
11: (msg, status) ← cpabe-dec (pub_key, priv_key, enc_msg)
12: **if** status = fail **then**
13: delete msg
14: **else**
15: show msg
16: **end if**
17: **end for**
18: last_refresh_TS ← Max (TS of all files)
19: **end procedure**

Pseudocode 6 Response to Revoke_List and AKV request

1: **procedure** PROCESS_REQUEST(' Revoke_List and Active_Key_Version '):
2: msg ← List of revoked user + AKV
3: signed_msg ← $(\text{msg})_{\text{KGC\_priv\_key}}$
4: Response (signed_msg)
5: **end procedure**

5 Security Analysis of the Model

The KGC, Repository and user exchange information between them over an unsecure network to achieve the functionalities described in Section 3.2. We justify that our model ensures Confidentiality, Integrity and Authentication of the information exchanged.

Any message that the KGC or Repository send to a user is signed by their private key. Signing ensures integrity of information as well as authentication of its source. This signing process takes place in two cases-

- When the user requests `Revoke_List` and `Active_Key_Version` from KGC, the KGC signs the packet containing this data before sending it to the user.
- When the user requests Repository for File Refresh,

Pseudocode 7 Response to attribute updation

Input:

user_id : user_id of User whose attributes have to be updated

new_attributes : Updated attributes of the user

1: **procedure** PROCESS_REQUEST('Attribute updation', user_id, new_attribute)
2: verify(new_attributes)
3: attributes[user_id] ← new_attributes
4: old_key_id ← key_id_map[user_id]
5: Request('Revoke Key', KGC, old_key_id)
6: key_id_map[user_id] ← new_keyID()
7: new_key ← cpabe-keygen (master_key, pub_key, attributes[user_id] + key_id_map[user_id] + AKV)
8: policy ← 'key_id = old_key_id'
9: $\{new\_key\}_{key\_id\ =\ old\_key\_id}$ ← cpabe-enc (pub_key, new_key, policy)
10: signed-pkt ← $(\{new\_key\}_{key\_id\ =\ old\_key\_id})_{KGC\_priv\_key}$
11: Response(signed-pkt)
12: **end procedure**

Pseudocode 8 Response to update key_version request

Input:

user_id : user_id of user who sends the Update Key_Version Request

AKV : Active_Key_Version of the system

1: **procedure** PROCESS_REQUEST ('Update Key_Version', user_id):
2: requester_key_id ← key_id_map[user_id]
3: new_key ← cpabe-keygen (master_key, pub_key, attributes[user_id] + requester_key_id + AKV)
4: policy ← 'key_id = requester_key_id'
5: $\{new\_key\}_{policy}$ ← cpabe-enc (pub_key, new_key, policy)
6: signed_pkt ← $(\{new\_key\}_{policy})_{KGC\_priv\_key}$
7: Response(signed-pkt)
8: **end procedure**

the Repository signs all the files before sending them to the user.

When confidential data is sent to any user, encryption is done using cpabe with appropriate policy. This ensures confidentiality of information. This takes place in two scenarios.

- When the KGC sends a renewed key to user, the KGC encrypts the renewed key using cpabe-enc. The policy specified is such that only the intended user's old key can decrypt it.
- When a user has to send files to other users, the user encrypts data using cpabe-enc before uploading it to Repository.

Pseudocode 9 Response to revoke user key request

Input:

key_id : key_id of key that has to be revoked

Revoke_List : List of key_ids that have been revoked

Threshold : Maximum allowed size of Revoke_List

1: **procedure** PROCESS_REQUEST('Revoke Key', key_id):
2: Revoke_List ← Revoke_List + key_id
3: **if** Size (Revoke_List) > Threshold **then**
4: AKV ← AKV + 1
5: Revoke_List ← ϕ
6: **end if**
7: **end procedure**

When a user wants to send any data to the Repository, an asymmetric key pair is generated by KGC when user wants to upload a file. Integrity of file is achieved because the file is signed by user before sending to Repository. Authentication is achieved because only the user who sent the 'Upload Request' will be able to decrypt K1 which is to be used to sign the data. Confidentiality of file is ensured because the file is encrypted using cp-abe before signing. This approach avoids the need for public-private key pair for each user.

5.1 Attack Scenarios

Figure 2 describes the messages that are exchanged between various entities. We provide an analysis of how our communication model remains secure in the case of attacks carried out by an adversary on each of these messages.

1) Adversary fabricates Message 1 and sends it to KGC - In this case, there is no issue because `Revoke_List` and `Active Key Version` are public information and our model doesn't require them to be held secretly.

2) Adversary intercepts Message 2, modifies it and sends the corrupted message instead - The attack will be curbed as Message 2 is signed by the KGC. If the adversary tries to carry out such an attack, the user will detect that the message has lost its integrity.

3) Adversary poses as a legitimate user U1 and tries to upload a harmful file to Repository (Message 3) - An authentication mechanism is used by the Repository to verify the uploader. The Repository sends a message containing a key using which the uploader

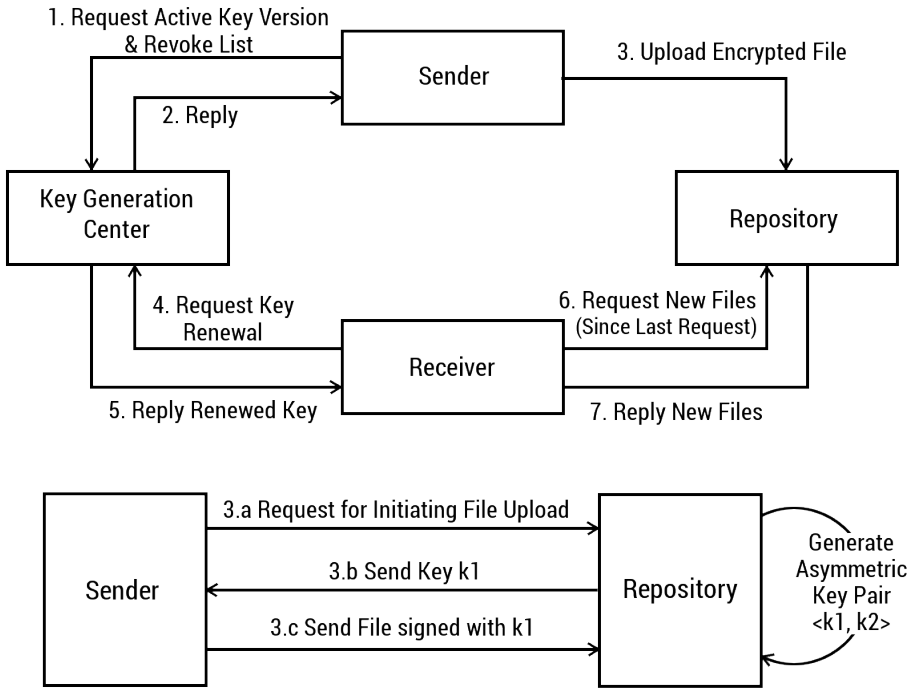


Figure 2 Communication between different entities

is supposed to sign his file. The Repository encrypts this message using cpabe-enc with policy such that only U1's key be able to decrypt it. Therefore, the adversary will not be able to extract the key from KGC's message and won't be able to sign the file.

4) A legitimate user perpetrates an insider attack by uploading a harmful file to Repository - Due to the authentication mechanism employed by the Repository, the uploader of each file is known. After detection of the harmful file, this information can be used to take action against the user.

5) Adversary fabricates Message 4 posing as a legitimate user U1 - The KGC encrypts the renewed key using cpabe-enc and a policy such that only U1's old key may be able to decrypt it. So, the adversary will not be able to decrypt the Message 5 from KGC. Therefore, the adversary will not be able to obtain U1's key. The only thing that this attack succeeds in doing is generating a futile response from KGC.

6) Adversary intercepts Message 5 from KGC to find out a user's key - Due to the same reason given in previous attack scenario, this attack won't work.

7) Adversary intercepts Message 7 that was being sent to user U1 OR Adversary sends Message 6 to Repository - The files uploaded on the Repository are encrypted using cpabe-enc. The Repository sends these encrypted files in response to a Message 6. The adversary will not be able to decrypt any files.

8) Adversary intercepts Message 7 that was being sent to user U1 and sends different files instead to U1 - The KGC signs the files before sending them to U1. Due to this, if an attacker tries to carry out such an attack U1 will detect that the files received by him are not from Repository.

In our proposed model, we rely on the security of the CP-ABE toolkit for the encryption and decryption process. We assume that data encrypted using the toolkit remains confidential. The toolkit implements the scheme proposed in [3].

6 Conclusion and Future Scope

The secure communication model proposed by us allows users to selectively share files among other users. It is more secure than using a server to enforce access control because in the event the Repository is compromised, our model ensures that the files would remain confidential. A user's key can be revoked, which effectively revokes all access rights of the user. The user's attributes can also be updated, which effectively changes his access rights.

As compared to a PKI-based approach, our model has the following advantages:

1) There is no need for managing multiple public keys using Certificate Authorities. There are only 3 public keys in our model: cpabe public key, KGC and Repository public key. These are available to all since the initial key distribution.

2) For sending file to N users only one encryption is required as opposed to N encryptions in case of PKI.

3) The sender can simply specify the attributes of the intended audience. As opposed to PKI, he doesn't need to know exactly who constitutes the intended audience. Due to this property of our model, there is no need for each User to store the list of all users along with their attributes.

These advantages hold as long as there exists a secure and scalable implementation of KGC.

The following are a few areas which can be worked upon to make our proposed model more secure and flexible:

1) Provide a mechanism to change the Master key-Public key pair of CP-ABE in case the Master key is compromised or a brute force attack is successful in discovering the Master key. It is a challenge to incorporate this functionality while still allowing operations on files encrypted before the Master key change.

2) Formalize the mechanism to change public-private key pair of KGC and Repository.

References

[1] S. G. Akl and P. D. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," *ACM Transactions on Computer Systems*, vol. 1, no. 3, pp. 239–248, 1983.

[2] J. Bethencourt, A. Sahai, B. Waters, *Ciphertext-Policy Attribute-Based Encryption*, Mar. 24, 2011. (`http://acsc.cs.utexas.edu/cpabe/`)

[3] J. Bethencourt, A. Sahai, B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy (SP'07)*, pp. 321–334, 2007.

[4] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography*, pp. 253–273, Springer, 2011.

[5] N. Chen and M. Gerla, "Dynamic attributes design in attribute based encryption," in *Annual Conference of ITA (ACITA)*, University of Maryland, MD, 2009.

[6] M. Chuah, S. Roy, and I. Stoev, "Secure descriptive message dissemination in dtns," in *Proceedings of the Second ACM International Workshop on Mobile Opportunistic Networking*, pp. 79–85, 2010.

[7] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[8] N. Doshi and D. Jinwala, "Updating attribute in cp-abe: A new approach.," *IACR Cryptology ePrint Archive*, vol. 2012, p. 496, 2012.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[10] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid, "Access control meets public key infrastructure, or: Assigning roles to strangers," in *Proceedings of IEEE Symposium on Security and Privacy (S&P'00)*, pp. 2–14, 2000.

[11] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Advances in Cryptology (EUROCRYPT'08)*, pp. 146–162, Springer, 2008.

[12] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal of Network Security*, vol. 15, no. 4, pp. 231–240, 2013.

[13] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proceedings of the 4th ACM International Symposium on Information, Computer, and Communications Security*, pp. 276–286, 2009.

[14] X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data," *International Journal of Network Security*, vol. 16, no. 6, pp. 437–443, 2014.

[15] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 99–112, 2006.

[16] R. L. Rivest, A. Shamir, and L. M. Adleman, *Cryptographic Communications System and Method*, US Patent 4,405,829, Sep. 20, 1983.

[17] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology (EUROCRYPT'05)*, pp. 457–473, Springer, 2005.

[18] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, pp. 47–53, Springer, 1985.

[19] S. G. Weber, "Securing first response coordination with dynamic attribute-based encryption," in *IEEE World Congress on Privacy, Security, Trust and the Management of e-Business (CONGRESS'09)*, pp. 58–69, 2009.

[20] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 261–270, 2010.

Jayam Modi, Manav Prajapati, Abhinav Sharma and Ravi Ojha obtained their B.Tech. in Computer Engineering in 2015 from Sardar Vallabhbhai National Institute of Technology, Surat, India. The work discussed here was a team effort towards the fulfilment of their B.Tech.

degree and was achieved under the guidance of Dr. Devesh Jinwala.

Devesh Jinwala is a Professor at Sardar Vallabhbhai National Institute of Technology, Surat, India. His major areas of interest are Information and Communications Security, Privacy and Cryptography, Security in Resource Constrained Devices, Software Requirements Specifications, Load Distribution and Failure Tolerance in Distributed Systems and Algorithms and Computational Complexity.

Turbo Unequal Error Protection Codes with Multiple Protection Levels

Qian Mao[1], and Chin-Chen Chang[2]
(Corresponding author: Chin-Chen Chang)

School of Optical-Electrical and Computer Engineering[1]
University of Shanghai for Science and Technology
No. 516, Jungong Rd., Yangpu, Shanghai, 200093, P. R. China
Department of Information Engineering and Computer Science[2]
Feng Chia University
No. 100, Wenhwa Rd., Seatwen, Taichung, 40724, Taiwan
(Email: alan3c@gmail.com)
(Received Sept. 30, 2015; revised and accepted Jan. 23 & Mar. 29, 2016)

Abstract

Most existing Turbo unequal error protection (UEP) codes provide two error protection levels at the cost of extending the block size of the codeword or increasing the computational complexity. This paper proposes two novel Turbo UEP coding schemes, which provide multiple error protection levels for information bits, according to their different sensitivities to the channel noise. By permuting the information bits and designing the puncturing scheme, multiple error protection levels can be achieved using one encoder without increasing the computational complexity. In addition, the coding rate of the proposed Turbo UEP schemes can be chosen from 1/3 to 1/2. Experimental results show that the information bits with a high protection level resist noise more effectively than those with a low protection level. The proposed UEP schemes provide better capability of error protection for the entire information, compared with the Turbo equal error protection (EEP) schemes and the existing Turbo UEP schemes.

Keywords: Bit Error Rate (BER), rate-compatible punctured convolutional (RCPC) codes, turbo codes, unequal error protection (UEP)

1 Introduction

Error correction coding is a popular measure that is used to resist channel noise in communication systems. There are many error correction codes, such as linear block code, convolutional code, and Turbo code. These codes have good error correction capabilities, however, they provide equal error correction capabilities to all the information bits even though information has different sensitivities to noise when transmitted in a communication channel. This diversity results from the different kinds of source information and the different bits that comprise a piece of information. Therefore, information bits should be protected to greater or lesser extents, depending on their sensitivities to the channel noise.

In order to provide variable error protection capabilities for different bits of information, several kinds of error correction coding schemes usually are used in a communication system [13, 20, 26]. In these systems, the important information is usually coded by some codes that have high error correction capabilities and low coding rates, while the less important information is coded by some other codes that have low error correction capabilities and high coding rates. Thus, information bits are provided different levels of error protection according to their different sensitivities to the noise.

However, using several coding schemes increases the complexity of the encoder and causes time delays. To provide unequal error correction capabilities by one encoder, Unequal Error Protection (UEP) codes were proposed. The UEP capabilities were achieved by improving the conventional error correction codes. By changing the structures of their coding space, the probability of the errors occurring in the important bits becomes less than the probability of their occurring in the less important bits after decoding. That is to say, the UEP codes usually don't decrease the overall number of errors, but control the locations at which the errors occur. By this approach, the UEP codes provide unequal error protections for information by a one-shot coding process and decrease the effect of channel noise on the entire information.

The UEP code was first proposed by Masnick and Wolf in 1967, and a linear block UEP code was proposed by them [17]. After that, many researchers made significant contributions in this field. Gils proposed a cyclic UEP

code and proved its capability of providing unequal error protection [9]. For the transmission of source messages that contain packets of different importance over lossy packet erasure links, Vukobratovic and Stankovic provided a performance analysis method of random linear UEP codes [8]. In order to achieve better error correction performance, some non-linear channel coding schemes were considered to achieve UEP capabilities. Hagenauer proposed a punctured convolutional coding scheme to obtain flexible coding rates to meet different error protection needs of the source information or different channel situations [10]. After that, the UEP coding schemes based on the rate-compatible punctured convolutional (RCPC) mechanism were studied extensively [21, 31].

In recent years, Turbo codes have been attracting more and more attention because of their perfect error correction capabilities. Turbo codes have been studied extensively and have been applied in a number of communication systems [3, 4, 7]. The conventional Turbo code provides equal error protection for every information bit. Since a Turbo encoder is composed of two recursive systematic convolutional (RSC) encoders, the UEP schemes for convolutional codes can be easily used in Turbo codes. Based on a rate-compatible puncturing mechanism, Barbulescu *et al*. proposed a Turbo UEP code with two error protection levels [2]. This scheme provided UEP capabilities for the information bits in a coding block, but it decreased the coding rate, compared to the conventional Turbo code. To overcome this problem, some special modulations or interleaving schemes must be used. Rowitch and Milstein proposed a hybrid forward-error correction/automatic repeat-request (FEC/ARQ) system which was based on Hagenauer's RCPC mechanism [22]. In addition, they proposed the criteria for designing the puncturing patterns in [23]. Jung and Plechinger proposed a design method for the rate-compatible punctured Turbo codes for mobile radio applications and illustrated its viability by simulation [12]. By partitioning the coding block of the Turbo code into many sub-blocks according to their importance, Caire and Biglieri achieved multiple UEP capability in a coding block [5]. Zhou improved Caire's scheme in [30]. However, for both schemes, the outputs of the two recursive systematic convolutional encoders were punctured independently, which decreases the average Bit Error Rate (BER) of the entire coding block. In addition, Aydinlik and Salehi derived the performance bounds of the Turbo UEP codes, which can be used to predict the codes' performance [1].

The UEP schemes can be used in image transmission to achieve better quality. Thomos et al. proposed an optimal UEP scheme for the compressed images, which employed Turbo codes and Reed-Solomon codes [25]. Lakhdar et al. proposed a UEP scheme, for which the puncturing operation was controlled by a periodic matrix [14]. They applied this Turbo UEP code in JPEG image transmission and achieved better image quality. Mao et al. proposed a Turbo UEP coding scheme and applied it for the transmission of images compressed by Discrete Cosine Transform (DCT) [16]. However, all of these schemes provide only two protection levels. Zhang et al. pointed out that each information bit in a Turbo block can possess a different protection level [28]. They proposed a Turbo code that provides descending protection capabilities for the information bits according to their locations in a block, and used this UEP scheme to JPEG image transmission to achieve better quality.

The Turbo UEP code decreases the BERs of the important bits at the cost of increasing the BERs of the unimportant bits, while the BER of the entire coding block varies very little. Therefore, the UEP capabilities should depend on the characteristics of the source information. That is to say, for an UEP coding scheme, the different sensitivities of the information bits to the channel noise should decide the number of the error protection levels, the length of each level, and the error correction capability of each level. But, until now, most Turbo UEP codes can provide only two protection levels, i.e., high and low error protection levels, and the error correction capability of each level is fixed and cannot be designed arbitrarily.

In this paper, we propose two novel Turbo UEP schemes. The main contributions of the proposed schemes are:

1) The proposed schemes provide multiple error protection levels by a one-shot coding process.

2) Most existing schemes obtain UEP capability at the cost of decreasing the coding rate. However, the proposed Turbo UEP codes have the same coding rate with the normal Turbo code.

The rest of the paper is organized as follows. Section 2 gives preliminary information about the proposed schemes. Section 3 shows the structures and algorithms of the proposed Turbo UEP codes. In Section 4, the performances of the proposed schemes are analyzed by simulations. Our conclusions are presented in Section 5.

2 Preliminaries

In this section, the coding theory of the conventional Turbo code, which is the foundation of the proposed Turbo UEP schemes, is briefly analyzed.

A Turbo encoder is composed of an interleaver, two RSC encoders, and a puncturing mechanism, as shown in Figure 1.

Assume that the generator matrix of both RSC encoders is $G(D) = [1 \quad g_2(D)/g_1(D)]$, where $g_1(D) = g_{10}+g_{11}D+\cdots+g_{1,K-1}D^{K-1}$, $g_2(D) = g_{20}+g_{21}D+\cdots+g_{2,K-1}D^{K-1}$, and K is the constraint length. Parameter $g_{i,j}(i = 1, 2$ and $j = 0, 1, \cdots, K-1)$ is a binary number that is pre-determined. In addition, we assume that the range of the input message of the encoder is $\{0, 1\}$, and d_k is the k^{th} information bit, where $k = 1, 2, \cdots, L$ and L is the size of the coding block. Then, for the input bit d_k, there are three output bits of the Turbo encoder, which

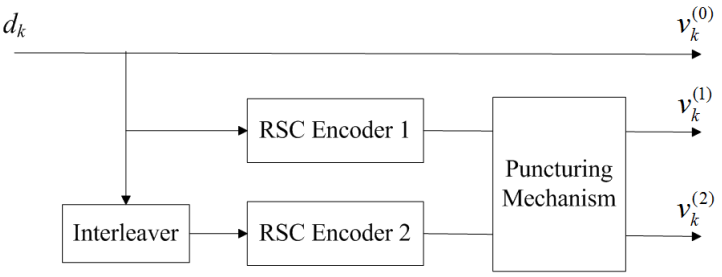


Figure 1: Framework of turbo encoder

are $v_k^{(0)}$, $v_k^{(1)}$, and $v_k^{(2)}$, as shown in Figure 1. The output bit $v_k^{(0)}$ is exactly the information bit d_k, i.e.,

$$v_k^{(0)} = d_k. \tag{1}$$

The second output $v_k^{(1)}$ is the parity-check bit of d_k computed by RSC Encoder 1, which is

$$v_k^{(1)} = \sum_{i=0}^{K-1} g_{2i} a_{k-i} \bmod 2, \tag{2}$$

where

$$\begin{aligned} a_r &= d_r + \sum_{i=1}^{K-1} g_{1i} a_{r-i} \bmod 2, \text{ and} \\ r &= k, k-1, \cdots, k-(K-1). \end{aligned} \tag{3}$$

To resist burst noise in the communication channel, the original bit sequence is randomly permuted by an interleaver, as shown in Figure 1. Then, the permuted bit sequence is sent to RSC Encoder 2, which has the same structure as RSC Encoder 1. By this approach, the third output $v_k^{(2)}$, which is also computed by Equations (2) and (3), is obtained [4].

Therefore, for each coding block, the output V of the Turbo encoder is composed of the original information bits and the parity-check bits, as shown in the following:

$$\begin{aligned} V = \{&v_1^{(0)}, v_1^{(1)}, v_1^{(2)}, v_2^{(0)}, v_2^{(1)}, v_2^{(2)}, v_3^{(0)}, v_3^{(1)}, \\ &v_3^{(2)}, v_4^{(0)}, v_4^{(1)}, v_4^{(2)}, \cdots, v_L^{(0)}, v_L^{(1)}, v_L^{(2)}\}. \end{aligned} \tag{4}$$

The coding rate (CR) of Equation (4) is 1/3. In order to enhance the coding rate, a puncturing mechanism can be used. Compared to parity-check bits, the information bits convey more information about the original message, thus, the puncturing operation only deletes parity-check bits. For the conventional Turbo encoder, the deleted bits are usually located periodically. For example, the puncturing algorithm may delete the bits on the even locations in $v_k^{(1)}$ and the bits on the odd locations in $v_k^{(2)}$. By this means, half of the parity-check bits are deleted, and the output $V_{\text{punctured}}$ with a size of $2L$ is obtained as follows:

$$\begin{aligned} V_{\text{punctured}} = \{&v_1^{(0)}, v_1^{(1)}, v_2^{(0)}, v_2^{(2)}, v_3^{(0)}, v_3^{(1)}, \\ &v_4^{(0)}, v_4^{(2)}, \cdots, v_L^{(0)}, v_L^{(2)}\}. \end{aligned} \tag{5}$$

Following that, the encoded sequence is first modulated to binary antipodal digits then is transmitted through a noisy channel. The sequence received by the recipient is denoted as:

$$\begin{aligned} R = \{&r_1^{(0)}, r_1^{(1)}, r_1^{(2)}, r_2^{(0)}, r_2^{(1)}, r_2^{(2)}, r_3^{(0)}, r_3^{(1)}, \\ &r_3^{(2)}, r_4^{(0)}, r_4^{(1)}, r_4^{(2)}, \cdots, r_L^{(0)}, r_L^{(1)}, r_L^{(2)}\}. \end{aligned} \tag{6}$$

In the receiver side, the decoder first uses a de-puncturing mechanism to classify $r_k^{(0)}$, $r_k^{(1)}$, and $r_k^{(2)}$ in R. The de-puncturing mechanism is the inverse operation of the puncturing mechanism of the encoder. After that, $r_k^{(0)}$ and $r_k^{(1)}$ are sent to Decoder 1, and $r_k^{(0)}$ and $r_k^{(2)}$ are sent to Decoder 2. Then, an iterative decoding process is conducted between the two decoders. The whole decoding process is shown as Figure 2.

In the decoding process, the two decoders implement a soft decision using the log-likelihood ratio of the received stream R. Assuming that there is a mapping for every transmitted bit $v_k^{(j)}(j = 0, 1, 2) : 0 \to -1$ and $1 \to +1$, then, for the Additive White Gaussian Noise (AWGN) channel, the log-likelihood ratio of the information bit d_k under the condition of $r_k^{(0)}$ is:

$$\begin{aligned} \lambda(d_k | r_k^{(0)}) &= \ln \frac{P(d_k = +1 | r_k^{(0)})}{P(d_k = -1 | r_k^{(0)})} \\ &= 4 \frac{E_s}{N_0} r_k^{(0)} + \ln \frac{P(d_k = +1)}{P(d_k = -1)} \\ &= \lambda_c r_k^{(0)} + \lambda_a(d_k), \end{aligned} \tag{7}$$

where E_s/N_0 is the Signal-to-Noise Ratio (SNR) of the channel, $\lambda_c = 4(E_s/N_0)$ is defined as the channel reliability factor, and $\lambda_a(d_k)$ is the *a priori* λ value of d_k.

For the parity-check bit $v_k^{(j)}(j = 1, 2)$, the log-likelihood ratio λ under the condition of $r_k^{(j)}$ is:

$$\lambda(v_k^{(j)} | r_k^{(j)}) = \lambda_c r_k^{(j)} + \lambda_a(v_k^{(j)}), \quad j = 1, 2. \tag{8}$$

When decoding, each decoder has three inputs, i.e., the soft outputs from the channel, which are $\lambda_c r_k^{(0)}$ and $\lambda_c r_k^{(1)}$(or $\lambda_c r_k^{(2)}$), and the *a priori* λ value $\lambda_a^{(1)}(d_k)$ (or $\lambda_a^{(2)}(d_k)$). For each decoder, the *a priori* λ value is the extrinsic information of another decoder, i.e., $\lambda_a^{(1)}(d_k) =$

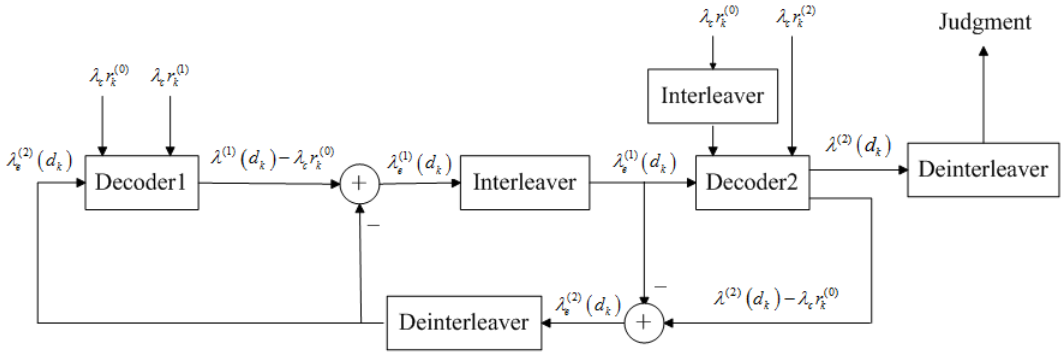


Figure 2: Framework of turbo decoder

$\lambda_e^{(2)}(d_k)$ and $\lambda_a^{(2)}(d_k) = \lambda_e^{(1)}(d_k)$. Each decoder has two outputs. The first one is $\lambda^{(1)}(d_k)$ (or $\lambda^{(2)}(d_k)$), which is the *a posteriori* λ value of d_k under the condition of the received bits and the *a priori* λ values. The second output is the extrinsic information of d_k, $\lambda_e^{(1)}(d_k)$(or $\lambda_e^{(2)}(d_k)$), which will be transferred to another decoder as the *a priori* λ value. Therefore, the two decoders implement an iterative, soft-decision algorithm. Each of iterations makes the judgment more reliable. The Turbo decoder outputs the final judgment after several iterations.

3 Proposed Turbo UEP Schemes with Multiple Protection Levels

The puncturing mechanism of the conventional Turbo code deletes half of the parity-check bits and enhances the coding rate from 1/3 to 1/2. This coding scheme provides Equal Error Protection (EEP) capabilities for all the information bits, which is defined as Turbo EEP code. In this section, two novel Turbo UEP schemes with multiple protection levels are proposed. The first UEP scheme has a flexible coding rate, and the second scheme has a fixed coding rate, which is 1/2.

3.1 Turbo UEP Code with Flexible Coding Rate

In digital communication, each bit for transmission usually has different importance for the transmitting contents. For example, the higher bits of 8-bits pixel value for an image are more important than the lower bits with respect to image representation. As a result, if the higher bits are damaged due to the channel noise, it will cause more serious influence on the transmitting contents than the condition that the lower bits are damaged. Therefore, in order to provide better protection for the bits with higher importance, we proposed a novel Turbo UEP coding scheme that provides different error correction capabilities to the information bits based on their importance to the transmitting contents.

Assuming that the block size of the Turbo UEP code is L and there are N protection levels in a block, of which protection capabilities decrease from the first level to the last level. Each protection level consists of $L_i (i = 1, 2, \cdots, N)$ information bits, thus, $L = L_1 + L_2 + \cdots + L_N$. Every information bit has two parity-check bits in a Turbo EEP coding scheme without puncturing. In our proposed UEP scheme, the number of the parity-check bits in each protection level is controlled in order to provide unequal protection level. That is to say, the higher the protection level is, the more information bits that have two parity-check bits there will be. We use the puncturing controller, P_{flexible}, to define this characteristic.

$$P_{\text{flexible}} = [p_1 \ p_2 \ \cdots \ p_N], \tag{9}$$

where $p_i (i = 1, 2, \cdots, N)$ is the percent of the information bits that have two parity-check bits in the i^{th} protection level, and $p_i \in [0, 1]$. That is to say, in the i^{th} protection level, there are $L_i \cdot p_i$ information bits with two parity-check bits, and the rest of the information bits in this level have only one parity-check bit. The value of $p_i (i = 1, 2, \cdots, N)$ depends on the sensitivity of the information bits in the i^{th} protection level to the channel noise. In addition, all of the information bits that have two parity-check bits should be chosen randomly in order to keep the error correction capability of the entire coding block at an acceptable level. The puncturing mechanism of the proposed Turbo UEP code is shown in Figure 3. In this figure, the shadowed blocks indicate the parity-check bits that are deleted by the puncturing mechanism. Figure 3 shows that there are more information bits that have two parity-check bits in the high protection level, and there are more information bits that have only one parity-check bit in the low protection level.

The coding procedures of our proposed Turbo UEP code with multiple protection levels and a flexible coding rate are listed below:

1) Permute the information bits in a coding block, i.e., put all the bits that belong to the same protection level together and array all the protection levels from high to low.

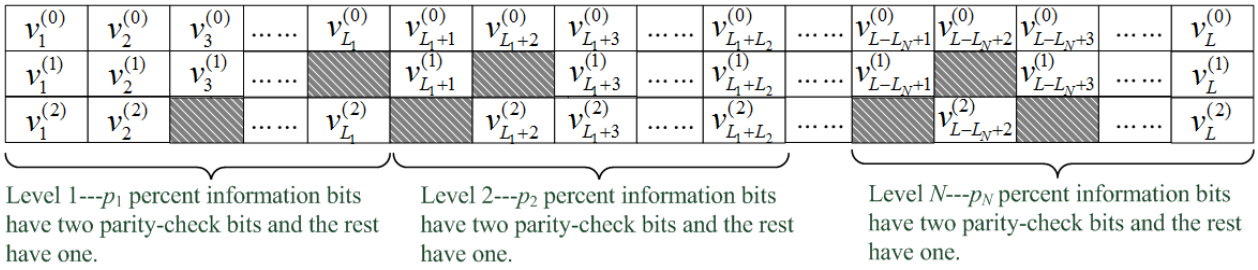


Figure 3: Puncturing mechanism of the turbo UEP code with multiple protection levels and a flexible coding rate

2) Determine the value of the puncturing controller, P_{flexible}, according to the different sensitivities to channel noise of the protection levels.

3) Design the puncturing mechanism for the Turbo encoder, according to P_{flexible}. For the i^{th} protection level, choose $L_i \cdot p_i$ information bits randomly, and reserve the two parity-check bits for them. The rest of the information bits in this protection level have only one parity-check bit. For the information bit $d_k(k = 1, 2, \cdots, L)$, if there should be only one parity-check bit and k is odd, $v_k^{(1)}$ is reserved for it; if k is even, $v_k^{(2)}$ is reserved.

4) Encode the rearranged information bits by (1) - (3) and get $v_k^{(0)}$, $v_k^{(1)}$, and $v_k^{(2)}$.

5) Puncture $v_k^{(0)}$, $v_k^{(1)}$, and $v_k^{(2)}$ by the proposed puncturing mechanism in Step 3. Transform the punctured outputs into a bit sequence and transmit it through the communication channel.

In the receiver side, the decoder uses the same puncturing algorithm to classify $r_k^{(0)}$, $r_k^{(1)}$, and $r_k^{(2)}$ in the received bit sequence, sends them to the decoders as shown in Figure 2, and starts an iterative decoding process. By this approach, the unequal error protection capabilities are achieved.

The coding rate of each protection level in the proposed Turbo UEP scheme is different, which depends on the value of p_i. The coding rate, CR_{flexible}, of the entire Turbo UEP code is:

$$CR_{\text{flexible}} = \frac{L}{\sum_{i=1}^{N}(L_i + L_i + L_i \cdot p_i)} = \frac{L}{\sum_{i=1}^{N} L_i(2 + P_i)}. \quad (10)$$

The information bits that correspond to higher importance levels are assigned with two parity-check bits, while the information bits that correspond to lower importance levels are assigned with only one parity-check bits. Therefore, the value of CR_{flexible} depends on the puncturing controller P_{flexible}, and $1/3 \leq CR_{\text{flexible}} \leq 1/2$. When CR_{flexible} is equal to $1/3$, all information bits are assigned with two parity-check bits, which means that there is no puncturing operation and the protection capability is equal to the Turbo EEP code without puncturing. When CR_{flexible} is $1/2$, all information bits are assigned with only one parity-check bit, which leads to a Turbo EEP code with a coding rate of $1/2$.

An example is shown in the following. Assuming that the original data are decimal numbers ranging from 0 to 255, then, each number can be denoted as an 8-bit byte. It is clear that the highest bit in a byte is the most important and most sensitive to channel noise, and every bit in a byte has a different sensitivity to channel noise. Therefore, the proposed Turbo UEP code partitions a coding block into eight protection levels. All the highest bits of the 8-bit bytes are provided the highest error protection, all the second-highest bits are provided the second-highest error protection, and so on. The puncturing controller, $P_{\text{flexible},1}$, in this scheme is:

$$P_{\text{flexible},1} = \left[1 \ \frac{1}{2} \ \frac{1}{2^2} \ \frac{1}{2^3} \ \frac{1}{2^4} \ \frac{1}{2^5} \ \frac{1}{2^6} \ \frac{1}{2^7}\right]. \quad (11)$$

Puncturing controller $P_{\text{flexible},1}$ means that all the information bits in the first protection level have two parity-check bits, half of the information bits in the second protection level have two parity-check bits, and so on. Therefore, the coding rate of the Turbo UEP code with puncturing controller $P_{\text{flexible},1}$ is

$$\begin{aligned} CR_{\text{flexible},1} &= \frac{L}{\sum_{i=1}^{8} \frac{L}{8}\left(2 + \frac{1}{2^{i-1}}\right)} \\ &= \frac{8}{16 + \left(1 + \frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^5} + \frac{1}{2^6} + \frac{1}{2^7}\right)} \\ &\approx 0.44. \end{aligned} \quad (12)$$

3.2 Turbo UEP Code with Fixed Coding Rate

The Turbo UEP scheme proposed in Section 3.1 has a flexible coding rate. In order to achieve a fixed coding rate for the Turbo UEP code with multiple protection levels, another puncturing mechanism is proposed in this section. In this scheme, some bits in the high protection levels have two parity-check bits, while some bits in the low protection levels do not have a parity-check bit.

The puncturing controller, P_{fixed}, for the Turbo UEP code with a fixed coding rate can be denoted as:

$$P_{\text{fixed}} = [p_1\ p_2\ \cdots\ p_N], \tag{13}$$

where $p_i \in [-1, 1]$ and $i = 1, 2, \cdots N$. For the i^{th} protection level, if p_i is positive, there should be $L_i \cdot p_i$ information bits that have two parity-check bits; if p_i is negative, there should be $L_i \cdot |p_i|$ information bits that do not have a parity-check bit, where $|x|$ denotes the absolute value of x. In order to make the coding rate 1/2, the following requirement should be satisfied:

$$\sum_{i=1}^{N} L_i p_i = 0,\ i = 1, 2, \cdots, N. \tag{14}$$

The value of $p_i (i = 1, 2, \cdots, N)$ in Equation (13) depends on the sensitivity of the information bits in the i^{th} protection level to channel noise. Both the bits that have two parity-check bits and the bits that do not have a parity-check bit should be chosen randomly. The puncturing mechanism of the Turbo UEP code with multiple protection levels and a fixed coding rate is shown in Figure 4.

The coding procedures of the Turbo UEP code with a fixed coding rate are listed below:

1) Permute the information bits in a coding block. Put all the bits that belong to the same protection level together, and array all the protection levels from high to low.

2) Determine the value of the puncturing controller, P_{fixed}, according to the sensitivities of the protection levels to channel noise.

3) Design the puncturing mechanism for the encoder, according to P_{fixed}. For the i^{th} protection level, if p_i is positive, choose $L_i \cdot p_i$ information bits randomly and reserve their two parity-check bits; if p_i is negative, choose $L_i \cdot |p_i|$ information bits randomly and delete both of their parity-check bits. The rest of the information bits in this protection level have only one parity-check bit. For the information bit $d_k (k = 1, 2, \cdots, L)$, if there is only one parity-check bit and k is odd, $v_k^{(1)}$ is reserved; if k is even, $v_k^{(2)}$ is reserved.

4) Encode the rearranged information bits by (1) - (3) and get $v_k^{(0)}$, $v_k^{(1)}$, and $v_k^{(2)}$.

5) Puncture $v_k^{(0)}$, $v_k^{(1)}$, and $v_k^{(2)}$ by the proposed puncturing mechanism in Step 3. Transform the punctured outputs into a bit sequence and transmit it through the communication channel.

An example is shown as follows. Assuming that the original data are decimal numbers and every number is denoted as an 8-bit byte, the information bits in a coding block are partitioned into eight levels. All the highest bits of the 8-bit bytes are arrayed in the first level, all the second-highest bits are arrayed in the second level, and so on. To provide higher protection for the important bits and to keep the coding rate of the entire block as 1/2, the puncturing controller can be designed as follows:

$$P_{\text{fixed},1} = [0.3\ 0.15\ 0\ 0\ -0.1\ -0.1\ -0.1\ -0.15]. \tag{15}$$

In this case, there are five protection levels in the UEP scheme and the coding rate is 1/2. The protection levels are designed as:

Level 1. The highest protection level. It consists of the highest bits of all the 8-bit bytes in a coding block. Among these highest bits, 30% of the information bits, which are chosen randomly, have two parity-check bits, and the rest of the information bits have only one parity-check bit each.

Level 2. The second-highest protection level. It consists of the second-highest bits of all the bytes in a block. Fifteen percent of the bits, which are chosen randomly, have two parity-check bits, and the rest of the bits have only one parity-check bit each.

Level 3. The third protection level. It consists of all the third and fourth bits of the 8-bit bytes, and all of the information bits in this part have only one parity-check bit.

Level 4. The fourth protection level. This is the second-lowest protection level, and it consists of all the fifth to the seventh bits of the bytes. Ten percent of the information bits in this protection level are chosen randomly, and they do not have a parity-check bit; the rest of the information bits have one parity-check bit each.

Level 5. The fifth protection level. This is the lowest protection level, and it consists of the lowest bits of all the bytes in a coding block. Fifteen percent of the information bits in this level are chosen randomly, and they have no parity-check bit; the rest of the information bits have one parity-check bit each.

Note that there are many reasonable schemes of the puncturing controller. Generally speaking, the number of the parity-check bits reserved in a protection level should be consistent with the importance of the information bits in this level. On the other hand, if too many information bits have no parity-check bit, the error correction performance of the entire coding block will be decreased. A large number of experiments show that the absolute value of $p_i (i = 1, 2, \cdots, N)$ in Equation (13) should not be larger than 50%. The value of P_{fixed} should be a trade-off between the UEP effect and the error correction performance of the entire coding block.

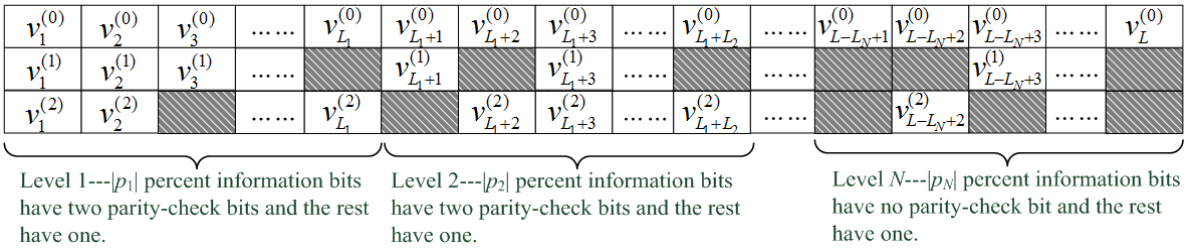


Figure 4: Puncturing mechanism of the turbo UEP code with multiple protection levels and a fixed coding rate

3.3 Evaluation Method of the Turbo UEP Code

The advantage of Turbo UEP code is that it provides unequal error correction capabilities for the information bits, according to their different sensitivities to channel noise. Therefore, the protection effect for the entire information is better than that of the Turbo EEP codes. In order to measure the different error correction capabilities for the information bits, the BER of each protection level should be measured and analyzed.

In order to measure the protection effect for the entire information, the standard deviation between the original data and the decoded data is used in this paper. Assuming that $X = [x_1, x_2, \cdots, x_Q]$ is the original information sequence in decimal form, $Y = [y_1, y_2, \cdots, y_Q]$is the decoded sequence in decimal form, and the lengths of sequences X and Y are both Q, the standard deviation, SD, between X and Y is:

$$SD = \left\{ \frac{\sum_{i=1}^{Q} [|x_i - y_i| - |\overline{X - Y}|]^2}{Q} \right\}^{\frac{1}{2}}, \tag{16}$$

where $|x_i - y_i|$ is the absolute value of $x_i - y_i$, and $|\overline{X - Y}|$ is the mean value of the absolute value of the difference between X and Y.

4 Experimental Results

In this section, the error correction performances of the proposed Turbo UEP codes are analyzed by simulation. All of the experiments were conducted in Matlab on a PC with 3.40 GHz Intel Core i7 CPU, 8GB main memory and Windows 7 OS. In the experiments, the transmission channel was the AWGN channel with Binary Phase Shift Keying (BPSK) modulation, and the input data were numbers ranging from 0 to 255. The parameters of Turbo code are shown in Table 1.

In the experiments, the 800 information bits in a coding block (excepting the two tail bits) were partitioned into eight protection levels, which are represented as PL_1 to PL_8 from the highest level to the lowest level. Each protection level consists of 100 information bits. Since each original number can be denoted as an 8-bit byte, in our experiments, all the highest bits in a coding block were provided the highest protection level (PL_1), all the second-highest bits were provided the second-highest protection level (PL_2), and so on. For comparison, the following two Turbo UEP schemes were also analyzed:

Table 1: Parameters of coding

Item	Value
Generate Matrix	$g_1(D) = 1 + D + D^2$ and $g_2(D) = 1 + D^2$
Decoding Algorithm	Log-MAP
Iteration Number	5
Block Length	802 bits
Quantity of the Protection Levels	8
Length of Each Protection Level	100 bits

- Turbo UEP scheme proposed by Z. D. Zhou [30]. In this scheme, the outputs of each RSC encoder are punctured by an independent puncturing matrix, which reserves all of the information bits and randomly punctures the parity-check bits according to the coding rate of each protection level.
- Turbo UEP scheme proposed by A. M. Lakhdar [14]. In this scheme, the outputs of the RSC encoders are punctured by a periodic puncturing matrix. In each period, all of the information bits are arrayed from high protection level to low protection level. For all of the information bits, 1) the information bits are reserved, and 2) the outputs of the second RSC encoder are alternatively punctured. The outputs of the first RSC encoder are punctured according to the protection level of the information bit. If the bit is highly protected, the first parity-check bit is reserved; if the bit is lowly protected, the first parity-check bit is alternatively punctured or completed punctured, according to the required coding rate.

In the following, the error protection performance of the proposed UEP Scheme 1 is first simulated. In the experiments, the SNR of the AWGN channel was 1.0 dB.

The experimental results are shown in Table 2. From the table, we see that the BER of the information bits with a high protection level is lower than that with a low protection level. This is because that there are more parity-check bits in this part. For comparison, Zhou's Turbo UEP scheme and a Turbo EEP scheme, which also have a coding rate of 0.44, are simulated. In Zhou's scheme, the parity-check bits of each RSC encoder are randomly punctured, keeping the coding rate of each protection level equal to the number indicated by Equation (11). In order to construct a Turbo EEP code with a coding rate of 0.44, some information bits are randomly chosen in the entire coding block and are reserved two parity-check bits, and the rest of the information bits have one parity-check bit. By this means, the coding rate of the Turbo EEP code can be controlled arbitrarily. From Table 2, we see that, although the average BER of the proposed UEP scheme is higher than that of the EEP scheme, which is due to the controlled puncturing mechanism of the UEP scheme, the standard deviation of the UEP scheme is lower than that of the EEP scheme. This means that the protection effect of the proposed UEP Scheme 1 is better than that of the EEP scheme with the same coding rate. For Zhou's Turbo UEP scheme, since the outputs of each RSC encoder are punctured independently, there are a quantity of information bits that have no parity-check bits, which leads to a highest BER and a highest standard deviation among the three schemes, as shown in Table 2.

The second experiment simulated a Turbo UEP scheme with a flexible coding rate and less protection levels. The puncturing controller of the proposed UEP Scheme 2 is:

$$P_{\text{flexible},2} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]. \tag{17}$$

From Equation (17), we see that there are only two different protection levels in this UEP scheme, i.e., all the highest bits are provided a high protection level, and all remaining bits are provided a low protection level. The coding rate, $CR_{\text{flexible},2}$, of this scheme is:

$$CR_{\text{flexible},2} = \frac{L}{\frac{L}{8}\cdot 3 + \frac{L}{8}\cdot 2\cdot 7} = \frac{8}{17} \approx 0.47. \tag{18}$$

The experimental results of the UEP Scheme 2 are shown in Table 2. From the table, we see that the BER of level PL_1 is lower than that of levels PL_2 to PL_8, and the last seven protection levels have approximately the same BER values. For the UEP scheme in [30], two independent puncturing matrices are generated, which reserves all of the parity-check bits for the high protection information bits and randomly punctures half of the parity-check bits of the low protection bits. For the UEP scheme in [14], since there is only one high protection bit in a period, the puncture matrix is:

$$P_{L1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

This puncturing matrix provides two protection levels for the eight information bits in a byte and repeats the UEP with a period of eight, and the coding rate is 0.47. Table 2 shows that although the average BER of the proposed Turbo UEP Scheme 2 is not the best one, the standard deviation of the proposed scheme is the lowest. This means that the proposed scheme provides the best protection for the entire information.

The third experiment simulated the protection effects of the proposed UEP schemes with a fixed coding rate, which is 1/2. The puncturing controller of Scheme 3 is shown as Equation (15). Therefore, there are five protection levels in this scheme. The experimental results are shown in Table 2. We can see that the higher the protection level is, the lower its BER becomes.

The puncturing controller of the proposed UEP Scheme 4 is:

$$P_{\text{fixed},2} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ -1]. \tag{19}$$

There are three protection levels in this scheme. All the highest bits are provided high protection, all the lowest bits are provided low protection, and the second through the seventh bits are provided middle error protection level. For comparison, the following three coding schemes are simulated: 1) a Turbo EEP code with a coding rate of 0.5 (EEP Scheme 3); 2) Zhou's UEP scheme [30], which reserves the two parity-check bits for the high protection bits, deletes all of the parity-check bits for the low protection bits, and randomly punctures half of the parity-check bits for the middle protection bits; and 3) Lakhdar's UEP scheme [14], which uses a puncturing matrix as follows:

$$P_{L2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

The coding rates of the five schemes are both 1/2. Table 2 shows that the lowest standard deviation is achieved by the proposed UEP Scheme 4.

From the above experiments, we see that the average BER of the entire information of our proposed UEP scheme is increased, compared to the Turbo EEP codes. This is because the puncturing controller of the UEP scheme lowers the chaos of the received bit stream, thereby reducing the decoding performances. But the standard deviation of the proposed UEP schemes was lower than those of the EEP schemes and the existing UEP schemes with the same coding rate. This means that the proposed UEP schemes have the best protection effects for the entire information.

Figures 5 through 7 show the comparisons of the standard deviation between the proposed Turbo UEP schemes and the existing schemes with varying SNR values. Figure 5 shows the protection effects of the proposed UEP Scheme 1, Zhou's UEP scheme [30], and EEP Scheme 1. The coding rates of them are 0.44. Figure 6 shows the protection effects of the proposed UEP Scheme 2, Zhou's UEP scheme [30], Lakhdar's UEP scheme [14], and EEP Scheme 2. The coding rates of them are 0.47. These experimental results show that the standard deviation of the

Table 2: BER performances (1.0 dB)

Coding Rate	0.44			0.47				0.5				
Scheme	EEP Scheme1	UEP [30]	Proposed Scheme1	EEP Scheme2	UEP [30]	UEP [14]	Proposed Scheme2	EEP Scheme3	UEP [30]	UEP [14]	Proposed Scheme3	Proposed Scheme4
BER of PL_1	/	8.47e-3	4.71e-3	/	9.98e-3	1.20e-2	5.70e-3	/	9.41e-3	1.89e-2	1.45e-2	8.04e-3
BER of PL_2	/	1.28e-2	8.96e-3	/	2.73e-2	1.42e-2	1.70e-2	/	3.34e-2	2.19e-2	1.90e-2	2.24e-2
BER of PL_3	/	1.83e-2	1.18e-2	/	2.74e-2	1.25e-2	1.89e-2	/	3.28e-2	2.00e-2	2.01e-2	2.85e-2
BER of PL_4	/	2.10e-2	1.31e-2	/	3.12e-2	1.41e-2	1.92e-2	/	3.23e-2	2.15e-2	2.06e-2	2.43e-2
BER of PL_5	/	2.11e-2	1.43e-2	/	2.79e-2	1.25e-2	1.86e-2	/	2.98e-2	1.92e-2	2.17e-2	2.74e-2
BER of PL_6	/	2.21e-2	1.48e-2	/	3.20e-2	1.44e-2	1.93e-2	/	2.98e-2	2.13e-2	2.26e-2	2.40e-2
BER of PL_7	/	2.23e-2	1.54e-2	/	3.15e-2	1.28e-2	1.88e-2	/	3.86e-2	1.95e-2	2.49e-2	2.54e-2
BER of PL_8	/	2.26e-2	1.56e-2	/	2.65e-2	1.41e-2	1.79e-2	/	6.35e-2	2.28e-2	2.50e-2	5.50e-2
Average BER	9.43e-3	1.86e-2	1.23e-2	1.37e-2	2.67e-2	1.33e-2	1.69e-2	1.79e-2	3.37e-2	2.07e-2	2.10e-2	2.69e-2
Standard Deviation	13.77	14.72	11.48	17.62	17.69	17.23	13.72	19.88	18.25	20.45	18.45	16.17

proposed UEP schemes is always lower than that of the existing UEP schemes and the EEP schemes. This means that for the entire information, the protection effects of the proposed Turbo UEP schemes with flexible coding rates are better than that of the existing UEP schemes and the EEP schemes with the same coding rate.

Figure 7 shows the protection effects of the proposed UEP Schemes 3 and 4, UEP schemes in [30] and [14], and EEP Scheme 3. The coding rates of these five schemes are 0.5. The experiments show that the UEP scheme in [30] has the highest standard deviation, and the standard deviation of the UEP scheme proposed in [14] is approximately the same as that of the EEP Scheme 3. For the proposed UEP Schemes 3 and 4, the standard deviations are lower than that of the EEP Scheme 3 and Lakhdar's UEP scheme when the SNR is less than 1.4 dB. But when the SNR increases, the standard deviation of the proposed UEP Scheme 3 is approximately the same as that of the EEP Scheme 3, while the standard deviation of the proposed UEP Scheme 4 is higher than that of the EEP Scheme 3. This means that the protection effects of the proposed Turbo UEP schemes with a fixed coding rate are better than that of the existing UEP schemes and the EEP scheme only when the channel noise is high.

The following experiments show the protection effects of the proposed scheme in image transmission. Assume that the gray values of pixels vary from 0 to 255. Since each bit of a pixel's gray value has different sensitivity to channel noise, the proposed UEP Scheme 1, which has eight different protection levels, are used in the following experiments. (The puncturing controller of UEP Scheme 1 is shown as Equation (11).) Figures 8 through 10 show the experimental results. In these figures, (a) is the original image, (b) is the decoded image using the proposed UEP Scheme 1, and (c) is the decoded image using EEP Scheme 1. The SNR of the transmission channel is 1.4 dB. Table 3 shows the Peak Signal-to-Noise Ratios (PSNR) and structural similarity (SSIM) values of the decoded images using different error protection

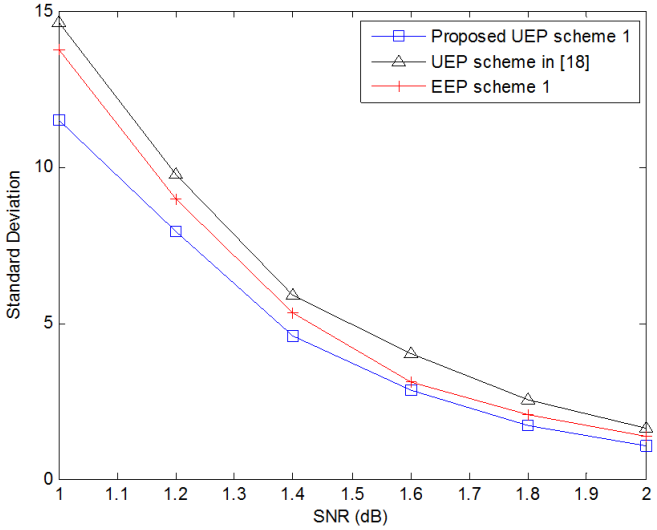


Figure 5: Standard deviation when coding rate = 0.44

schemes. The experimental results show that although UEP Scheme 1 and EEP Scheme 1 have the same coding rate, the PSNRs of the decoded images were increased about 2 dB by the proposed UEP scheme. Meanwhile, the SSIM values of the proposed UEP scheme are higher than those of the EEP scheme, which means that the proposed UEP scheme provides better visual quality of the reconstructed images.

5 Conclusions

Two Turbo UEP schemes were proposed in this paper, both of which provide multiple protection levels for information by a one-shot coding process. In the proposed UEP schemes, the entire coding block is partitioned into several protection levels, and the coding rate of each level is controlled independently. Simulations show that for both of the proposed UEP schemes, the information bits

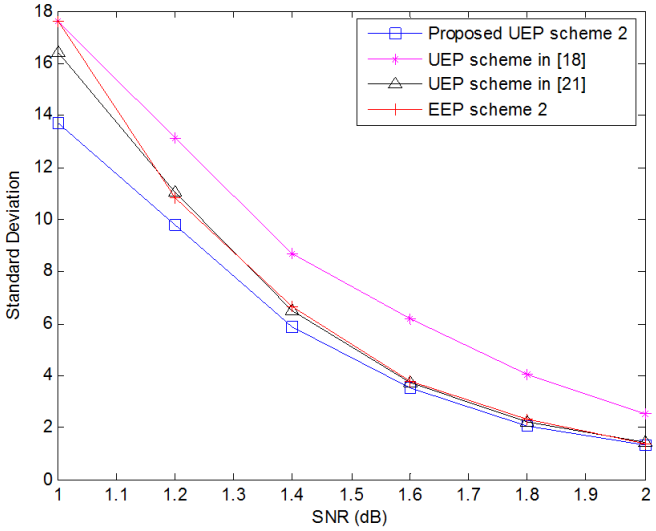


Figure 6: Standard deviation when coding rate = 0.47

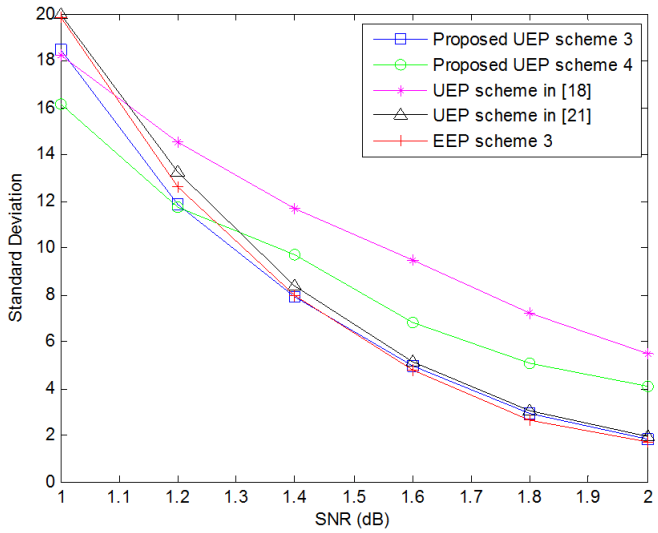


Figure 7: Standard deviation when coding rate = 0.5

Table 3: PSNRs of the decoded images using different protection schemes

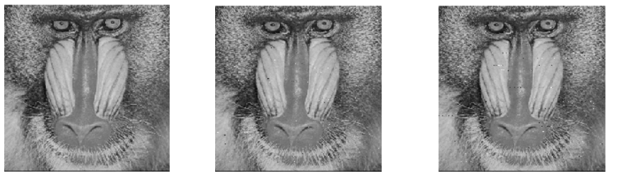
	UEP Scheme 1		EEP Scheme 1	
Cover Image	PSNR (dB)	SSIM	PSNR (dB)	SSIM
Parrot	34.36	0.9494	32.97	0.9440
Lena	34.01	0.9542	32.04	0.9449
Baboon	34.17	0.9762	32.06	0.9667

(a) Original Image (b)Decoded Image Using UEP Scheme 1 (c)Decoded Image Using EEP Scheme 1

Figure 8: Experimental results — Parrot

(a) Original Image (b)Decoded Image Using UEP Scheme 1 (c)Decoded Image Using EEP Scheme 1

Figure 9: Experimental results — Lena



(a) Original Image (b)Decoded Image Using UEP Scheme 1 (c)Decoded Image Using EEP Scheme 1

Figure 10: Experimental results — Baboon

with a higher protection level have a lower BER than those with a lower protection level. The first proposed UEP scheme has a flexible coding rate, which is more than 1/3 and less than 1/2. The protection effect for the entire information of this scheme is always better than that of the existing Turbo UEP schemes and the Turbo EEP scheme with the same coding rate. The second proposed UEP scheme has a fixed coding rate, which is 1/2. The protection effect of this scheme is better than that of the existing UEP schemes and the EEP scheme with the same coding rate when the channel noise is high. Further works may focus on the applications of Turbo UEP schemes [6, 11, 15, 18, 19, 24, 27, 29] in different kinds of source information.

References

[1] M. Aydinlik and M. Salehi, "Performance bounds for unequal error protecting turbo codes," *IEEE Transactions on Communications*, vol. 57, no. 5, pp. 1215–1220, 2009.

[2] A. S. Barbulescu and S. S. Pietrobon, "Rate compatible turbo codes," *Electronics Letters*, vol. 31, no. 7, pp. 535–536, 1995.

[3] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 409–428, 1996.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo codes," in *Proceedings of IEEE International Conference on Communication*, pp. 1064–1070, 1993.

[5] G. Caire and E. Biglieri, "Parallel concatenated codes with unequal error protection," *IEEE Transactions on Communications*, vol. 45, no. 5, pp. 565–567, 1998.

[6] R. Y. Chang, S. J. Lin, and W. H. Chung, "A method for the construction of hierarchical generalized space shift keying (gssk) modulation for unequal error protection," *Physical Communication*, vol. 9, pp. 88–96, 2013.

[7] I. Chatzigeorgiou, M. R. D. Rodrigues, I. J. Wassell, and R. A. Carrasco, "Analysis and design of punctured rate-1/2 turbo codes exhibiting low error floors," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 6, pp. 944–953, 2009.

[8] V. Stankovic D. Vukobratovic, "Unequal error protection random linear coding strategies for erasure channels," *IEEE Transactions on Communications*, vol. 60, pp. 1243–1252, May 2012.

[9] W. J. V. Gils, "Two topics on linear unequal error protection codes: Bounds on their length and cyclic code classes," *IEEE Transactions on Information Theory*, vol. IT-29, no. 6, pp. 866–876, 1983.

[10] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Communications*, vol. 36, pp. 389–400, Apr. 1988.

[11] J. 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.

[12] P. Jung and J. Plechinger, "Performance of rate compatible punctured turbo-codes for mobile radio applications," *Electronics Letters*, vol. 33, no. 25, pp. 2102–2103, 1997.

[13] K. Kang and W. J. Jeon, "Differentiated protection of video layers to improve perceived quality," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 292–304, Feb. 2012.

[14] A. M. Lakhdar, R. Meliani, and M. Kandouci, "Research on unequal error protection with punctured turbo codes in jpeg image transmission system," *Serbian Journal of Electrical Engineering*, vol. 4, pp. 95–108, June 2007.

[15] S. Li, Y. Zhao, L. He, Z. Wu, and Y. Li, "A novel stbc-dcsk transmission scheme for scalable video with unequal error protection property," in *Proceedings of 2016 IEEE International Conference on Consumer Electronics*, pp. 339–340, 2016.

[16] Q. Mao, B. Q. Xu, and Y. P. Qin, "A new scheme to improve the quality of compressed image transmission by turbo unequal error protection codes," in *Proceedings of the Seventh International Conference on Intelligent Information Hiding and Multimedia Signal (IIH-MSP'11)*, pp. 226–229, 2011.

[17] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. IT-3, no. 4, pp. 600–607, 1967.

[18] A. Naghdinezhadm and F. Labeau, "Frame distortion estimation for unequal error protection methods in scalable video coding," *Signal Processing: Image Communication*, vol. 29, no. 9, pp. 971–986, 2014.

[19] E. Namjoo, A. Aghagolzadeh, and J. Museviniya, "A new rateless code with unequal error protection property," *Computers & Electrical Engineering*, vol. 39, no. 7, pp. 1980–1992, 2013.

[20] A. Nosratinia, J. Lu, and B. Aazhang, "Source-channel rate allocation for progressive transmission of image," *IEEE Transactions on Communications*, vol. 51, pp. 186–196, Feb. 2003.

[21] F. Rey, M. Lamarca, and G. Vazquez, "Adaptive interleaver based on rate-compatible punctured convolutional codes," *IEEE Transactions on Communications*, vol. 57, pp. 1–6, June 2009.

[22] D. N. Rowitch and L. B. Milstein, "Rate compatible punctured turbo (rcpt) codes in a hybrid fec/arq system," in *Proceedings of IEEE Communication Theory Mini-Conference*, p. 55–59, Nov. 1997.

[23] D. N. Rowitch and L. B. Milstein, "On the performance of hybrid fec/arq systems using rate compatible punctured turbo (RCPT) codes," *IEEE Transactions on Communications*, vol. 48, no. 6, p. 948–959, 2000.

[24] M. Stanek, "A note on security protocol for multicast communications," *International Journal of Network Security*, vol. 14, no. 1, pp. 59–60, 2012.

[25] N. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Wireless image transmission using turbo codes and optimal unequal error protection," *IEEE Transaction on Image Processing*, vol. 14, pp. 1890–1901, Nov. 2005.

[26] H. X. Wang, C. T. Zhu, C. Y. Xiong, and S. P. Chen, "An effective joint source-channel coding with unequal error protection using asymmetric turbo codes," in *Proc. 14th International Conference on Advanced Communication Technology (ICACT'12)*, pp. 1006–1010, Feb. 2012.

[27] T. C. Yeh, J. R. Peng, S. S. Wang, and J. P. Hsu, "Securing bluetooth communications," *International Journal of Network Security*, vol. 14, no. 4, pp. 229–235, 2012.

[28] W. Zhang, X. Shao, M. Torki A. HajShirMohammadi, and I. V. Bajic, "Unequal error protection of jpeg2000 images using short block length turbo codes," *IEEE Communications Letters*, vol. 15, no. 6, pp. 659–661, 2011.

[29] Y. Zhang, K. Zhang, Y. Kang, and D. Yang, "Unequal error protection scheme for layered sources transmission over mimo systems using spatial diversity and multiplexing technology," *Journal of China Universities of Posts and Telecommunications*, vol. 22, no. 3, pp. 56–63, 2015.

[30] Z. D. Zhou and C. Xu, "An improved unequal error protection turbo codes," in *Proceedings of 2005 International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 284–287, 2005.

[31] D. C. Zou, X. C. Lu, H. T. Wu, and J. S. Xu, "Research on rate compatible punctured convolutional codes technique in OFDM-UWB indoor positioning system," in *Proceedings of the 2008 Congress on Image and Signal Processing (CISP'08)*, vol. 2, pp. 225–229, May 2008.

Qian Mao was born in Shanxi Province, China, in 1978. She received the B.S. degree in Mechanical Engineering and Automation Science from Nanjing University of Aeronautics and Astronautics, Jiangsu, China, in 2000, and M.E. degrees in Traffic Information Engineering and Control from Shanghai Ship and Shipping Research Institute, Shanghai, China, in 2003, and the Ph.D. degree in Traffic Information Engineering and Control from Tongji University, Shanghai, China, in 2006. Since 2006, she has been with the faculty of the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, where she is currently a Lecturer. She is also a post-doctoral researcher of Asia University, Taiwan. Her research interests include information theory and coding.

Chin Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from February 2005. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And, since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression, and data structures.

Two-phase Commit with Security Services: Using Distinctive Proofs to Relieve Fragile Communications

Yang Sun[1], Xueshuai Feng[2], Hongfeng Zhu[3]
(Corresponding author: Hongfeng Zhu)

Software College, Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang 110034, P.R. China
(Email: zhuhongfeng1978@163.com)
(Received July 2, 2015; revised and accepted Apr. 17 & Apr. 25, 2016)

Abstract

Inspired by stand-alone authentication, which can authenticate users when the connection to the central server is down, we present concepts called local proof and delayed proof that can adapt to two scenes when the authentication server is down: the former can solve self-authenticated to make local applications running without online authentication; the latter can solve two nodes to produce a session key for doing some transactions, but firstly they must exchange the delayed proof to prevent the fraud actions, specially, if the transaction is taking on the important process (such as contract signing or cash transaction), they must wait the authentication server is online. The key idea of our scheme is to improve the efficiency, and anyone can make effective use of the time to negotiate or do some unimportant things during the authentication server is down. Next, we propose a novel Chaotic Maps-based scheme against fragile communications, named CMFC, aiming to bypass the crashed authentication server temporarily for kinds of applications running. For important applications, we adopt the idea of two-phase commit protocol in our scheme: (1) the unavailable authentication server case, in which the CMFC can self-authenticated or compute a provisional delayed proof and a session key for two-party communicating. (2) the available authentication server case, in which, based on the phase (1) and the authentication server's verification, the two-party decides whether to commit (only if all have voted "Yes") or abort the transaction (otherwise). Finally, we give the formal security proof about our scheme with BAN logic and efficiency analysis.

Keywords: Ban logic; chaotic maps, delayed proof; stand-alone authentication

1 Introduction

In the arena of the network, a ubiquitous Internet demands pervasive connections, while keeping a stable communication environment is the foundation of our entire network. It is well known that user authentication mechanism is an essential stage for creating secure information systems, which can deter the spurious devices and services effectively. However, sometimes user authentication has to draw support from a remote central authorization server, namely, the user send a request to the central server firstly, then the server make a response to the user who requires authenticated, finally according to this challenge the user prove his identity. Each user needs to experience these steps for authentication. But from the whole process of authentication we can see that the central server is critical for completing authentication in the system. At the same time, it is also clear that due to the central server may suffer from various attacks and deliberate destructions, hence holding a dependable information system is not an easy task. Once the server under attack, the users will unable conduct authentication via the authorization server correctly, indicating the users only to wait in place until the server is available. In the long run, this situation will affect communication efficiency seriously and leave numerous hidden troubles for safety.

Yang et al. [14] present a two-factor (smart-card and password) mutual authentication protocol in 2008, aiming at build a generic construction framework for user authentication. Subsequently, in 2011, several researchers [5] put forward a new approach for authentication clients by three-factor (smart-card, password and biometrics). All of these stack factors are simply improving users vali-

date identities, whereas the above schemes are considering nothing about the dangerous that hidden in central server. With the purpose of achieving reliable communication between users and the central server, in 2014 Huang et al. [6] proposed a multi-factor authentication scheme for fragile communications. In this scheme, they first present a stand-alone authentication protocol that can authenticate users when the central server invalidation. Influenced by the stand-alone authentication mechanism, in this paper we present an efficient protocol to build a stable communication environment, which depends on local proof and delayed proof to complete users authentication when the central authorization server breakdown. Local proof is generated by the central server which can meet the requirement of users self-authentication [4] instead of online-authentication. Delayed proof and session key can help the users to set up a temporary platform to confer some inconclusive issues in the course of the authorization server is collapsed. But it is worth noting that the delayed proof is generated by proof, and only in the case of the central server unavailable can the two users employ the delayed proof in exchange. Meanwhile, if involves some crucial problems in the transmission, the users are not allowed to use the delayed proof, which means they must conduct this important process under the central server online. For example, in the process of online shopping, the buyer and the seller denote the two-node respectively, if the central server who refers to the AliPay collapse, two nodes can consult with each other privately with their own delayed proof and session key, but payment step must wait until the server (Alipay) recovery.

What we can see on previous works [2, 5, 10, 14] is that they merely provide the users authentication, but did not consider the situation of server crash. Aiming at meet the aforementioned requirements we first present a chaotic map-based scheme resist fragile communications [3], which we named Chaotic Maps-based scheme against fragile communications (CMFC) in the following scenario. Supposing there are two-node want to transmission under the authentication server called Alice and Bob, our scheme can be divided into two-stage roughly. The first stage we called self-authentication stage: When the authentication server is unavailable, Alice and Bob will build the delayed proof and session key by proof separately and to confer some unimportant affairs with their session key. The second stage named server authentication: Until the central server comes back online, Alice and Bob submit their own delayed proof to conduct authentication and if two-node successfully pass the certification finally, the server will permit the following communications between Alice and Bob.

In this paper, we design delayed proof framework and use the framework and chaotic maps [1, 8, 12, 15] to design the schemes for relieving fragile communications. We proposed several novel protocols which mainly intends to offer a temporary calculating platform to alleviate the waiting time of users. Therefore it is worth mentioning that after authenticated by the central server, the two-node have the right to determine whether they will continue to perform this service or opt-out, and only two-node all choose to continue is the process will keep on. From the above, you can see that our CMFC scheme is quite practical and necessary in critical situation when the server is unavailable because of certain attacks and destructions.

The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Next, a new Local Proof framework and two instances are described in Section 3. In Section 4, we describe delayed proof framework and an instance. In Section 5, we give the security of our proposed protocol. The efficiency analysis of our proposed protocol is given in Section 6. This paper is finally concluded in Section 7.

2 Chebyshev Chaotic Maps

Let n be an integer and let x be a variable with the interval [-1, 1]. The Chebyshev polynomial [13] $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ is defined as $T_n(x) = cos(ncos^{-1}(x))$. Chebyshev polynomial map $T_n : R \rightarrow R$ of degree n is defined using the following recurrent relation:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x),$$

where $n \geq 2$, $T_0(x) = 1$, and $T_1(x) = x$.

The first few Chebyshev polynomials are:

$$\begin{aligned} T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1, \\ &\vdots \quad\quad \vdots \end{aligned}$$

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that

$$T_r(T_s(x)) = T_{rs}(x).$$

An immediate consequence of this property is that Chebyshev polynomials commute under composition

$$T_r(T_s(x)) = T_s(T_r(x)).$$

In order to enhance the security, Zhang [9] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$. The enhanced Chebyshev polynomials are used in the proposed protocol:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x))(modN),$$

where $n \geq 2, x \in (-\infty, +\infty)$, and N is a large prime number. Obviously,

$$T_{rs}(x) = T_r(T_s(x)) = T_s(T_r(x)).$$

Definition 1. *(Enhanced Chebyshev polynomials) The enhanced Chebyshev maps of degree $n(n \in N)$ are defined as: $T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x))(mod\ p)$, where $n \geq 2$, $x \in (-\infty, +\infty)$, and p is a large prime number. Obviously, $T_{rs}(x) = T_r(T_s(x)) = T_s(T_r(x))$.*

Definition 2. *(DLP, Discrete Logarithm Problem) Given an integer a, find the integer r, such that $T_r(x) = a$.*

Definition 3. *(CDH, Computational Diffie-Hellman Problem) Given an integer x, and the values of $T_r(x), T_s(x)$, what is the value of $T_{rs}(x) \stackrel{?}{=} T_r(x)T_s(x)$*

It is widely believed that there is no polynomial time algorithm to solve DLP, CDH with a non-negligible probability.

3 Local Proof Framework and Instances

In this section, we first present a novel stand-alone authentication framework and two instances including password-based and biometric-password-based.

3.1 Notations

The concrete notations used hereafter are shown in Table 1. These notations can be used in Section 3 and Section 4.

3.2 Local Proof Framework

As we consider the efficiency and eliminate the verifier table on the server's side, we adopt only the server has the public key and secret key $\{(x, T_k(x)), K\}$. The server will compute the covered proof and local proof, and send them to the user by secure channel. Figure 1 illustrates the framework of the local authentication and some definitions are described as follows.

Definition 4. *Proof.*

A proof means that the server can compute it by his secret key to authenticate the specified user but the proof must be covered by password or biometric for preventing stored in the smart device directly.

Definition 5. *Covered Proof.*

A covered proof is a protection mechanism which must receive at least an external input (password or/and biometric) for recovering the proof.

Definition 6. *Local Proof.*

A local proof aims at adapting to stand-alone authentication when the authentication server is unavailable. In other words, the result of temporary computation by inputting the password or/and biometric which must equal to the local proof.

Remark 1. *The server is available means that both the communications and the authentication server are usable. The server is unavailable means that both the communications and the authentication server are unusable or either of them is unusable.*

3.3 Instance of Using Password

Figure 2 illustrates the instance of using password.

Step 1. When a user Alice wants to be a new legal user, she chooses her identity ID_A, password PW_A and a random number R. Then Alice computes $H(PW_A||R)$ and sends $\{ID_A, H(PW_A||R)\}$ to the server via a secure channel.

Step 2. Upon receiving $\{ID_A, H(PW_A||R)\}$ from the Alice, the server computes $P = H(ID_A||K)$ as the proof of the user, $V = H(PW_A||R) \oplus H(ID_A||K)$ as the covered proof, $L = H(H(ID_A||K))$ as the local proof and sends $\{V, L\}$ to the Alice.

Step 3. Alice stores $\{V, L, R\}$ securely. Storage carrier may be smart card, applications' database or others.

Local Authentication. Alice inputs password to recover $P = H(PW_A||R) \oplus V$. Then Alice compares $H(P)$ with L. Authentication succeeds only if the hash value matches with the local proof L.

3.4 Instance of Using Biometric and Password

Figure 3 illustrates the instance of using biometric and password.

Step 1. When a user Bob wants to be a new legal user, he chooses his identity ID_B, password PW_B, a random number R with inputting biometric image sample B. Then Bob computes $H(PW_A||R)$ and sends $\{ID_B, H(PW_B||R, B)$ to the server via a secure channel.

Step 2. Upon receiving $\{ID_B, H(PW_B||R, B)$ from the Bob, the server computes $P = H(ID_B||K)$ as the proof of the user, $V = H(PW_B||R) \oplus H(ID_B||K)$ as the covered proof, $L = H(PW_B||R) \oplus B$ as the local proof and sends $\{V, L\}$ to the Bob.

Step 3. Bob stores $\{V, L, R, \tau, d()\}$ securely.

Local Authentication. Bob inputs password to recover $B = H(PW_B||R) \oplus L$. Then Bob inputs B^* and verify $d(B^*, B) < \tau$? Authentication succeeds only if $d(B^*, B) < \tau$.

Table 1: Notations

Symbol	Definition
ID_i PW_i	The identity of user, the password of user, respectively
R, a, b	Nonces
$(x, T_K(x))$	The public key of the authentication server based on Chebyshev chaotic maps
K	The public key of the authentication server based on Chebyshev chaotic maps
B	The biometric sample of user
τ	Predetermined threshold for biometric verification
$d()$	The symmetric parametric function
H	A secure one-way hash function
$\|\|$	Concatenation operation

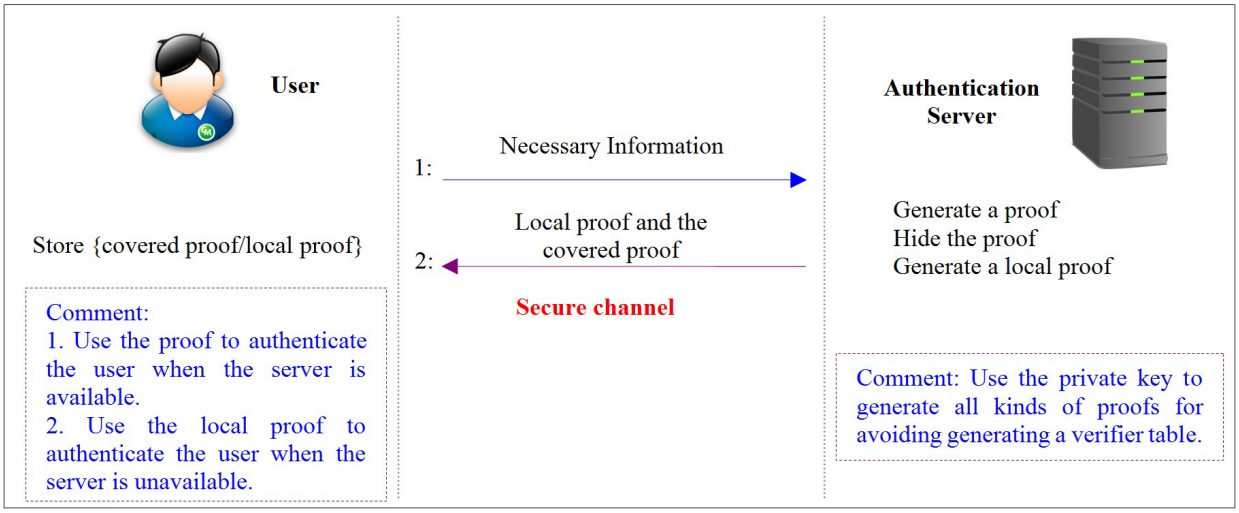


Figure 1: The framework of the local authentication

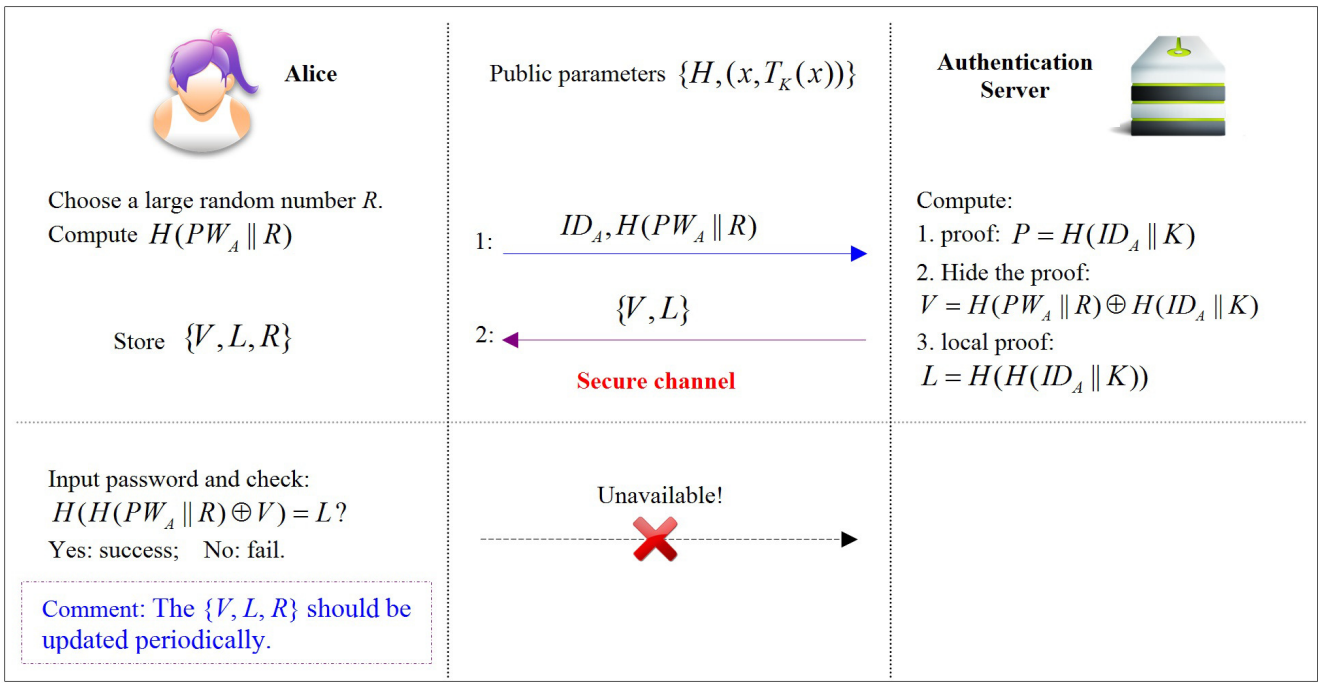


Figure 2: Chaotic maps-based for local authentication scheme with password

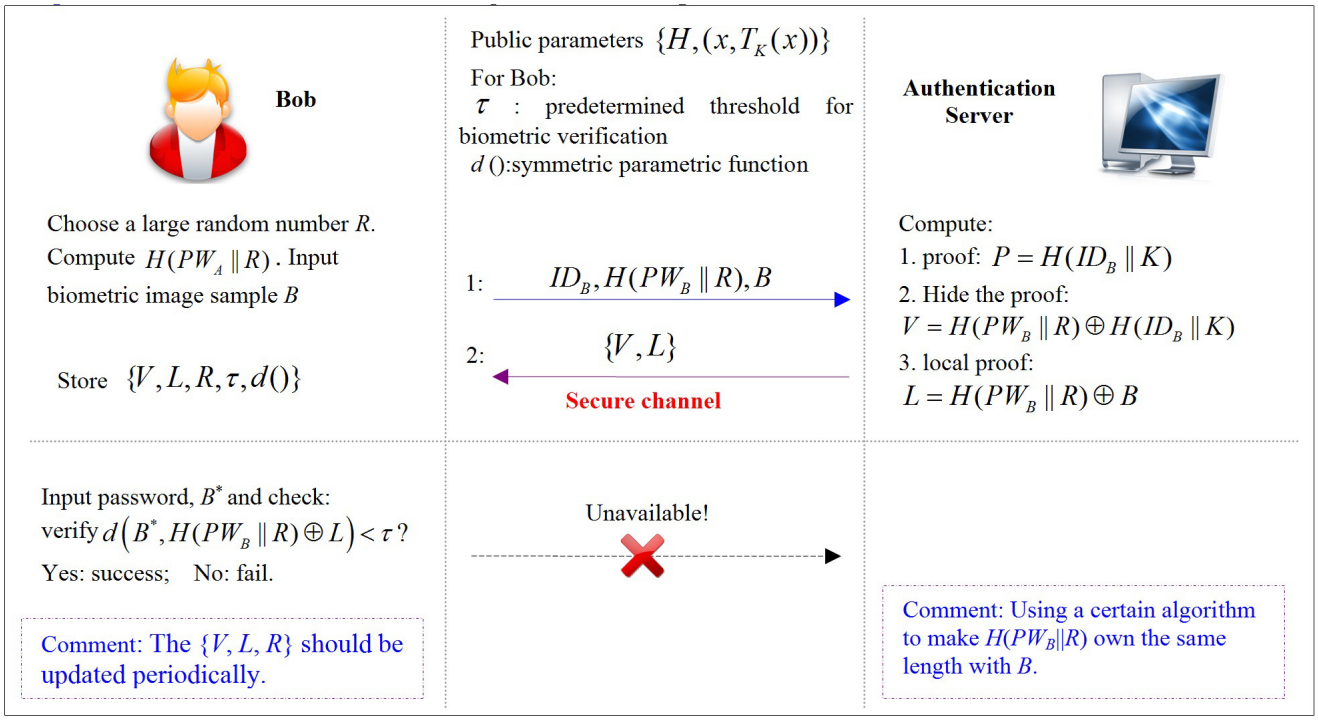


Figure 3: Chaotic maps-based for local authentication scheme with two-factor

4 Delayed Proof Framework and an Instance

In this section, we first present a novel Chaotic Maps-based local and delayed proof scheme based on password which is made up of three phases: registration phase, local authentication and delayed authentication.

In brief, the registration phase and local authentication phase are described as shown in Section 3.3 and Figure 2. So, in Section 4, we mainly describe the Delayed Proof Framework and an instance with password-based.

4.1 Delayed Proof Framework

The delayed proof is generated from proof which is covered by password or something else. So, the delayed proof is a temporary and random large number that can be used only once. Figure 4 illustrates the framework of the delayed authentication.

The first phase commit with security services (Generate delayed proof with a session key).

When two users (Alice and Bob) want to communicate with each other but the authentication server is unavailable. In order to accelerate efficiency and processing speed, they must temporary consult some unimportant things. So, they construct the delayed proof and session key at the same time based on the proof which can be authenticated by authentication server later.

The second phase commit with security services (Delayed Authentication).

When the two users want to do some important things and the authentication server is available, they submit the other's delayed proof to the server. After receiving the confirm response from the authentication server, the two users must authenticate the authentication server firstly, and then they will continue the unfinished work.

4.2 An Instance with Password-based

Figure 5 illustrates an instance with password-based of the delayed authentication.

The first phase commit with security services (Generate delayed proof with a session key).

Step 1. Alice inputs PW_A to get the proof: $H(ID_A||K)$. Then Alice selects a large random integer a and computes $T_a(x)$. Finally Alice computes delayed proof: $m_A = \{ID_A, T_a(x), H(ID_A||T_a(x)), DP_A = H(H(ID_A||K) \| T_a(x) \| ID_A)\}$ and sends it to Bob. The same way for Bob.

Step 2. Upon receiving $m_A = \{ID_A, T_a(x), H(ID_A||T_a(x)), DP_A = H(H(ID_A||K) \| T_a(x) \| ID_A)\}$ from Alice, Bob computes $H' = H(ID_A||T_a(x))$ and check if it equals to the received hash value $H(ID_A||T_a(x))$. If holds, Bob computes the session key $H(T_bT_a(x))$. The same way for Alice.

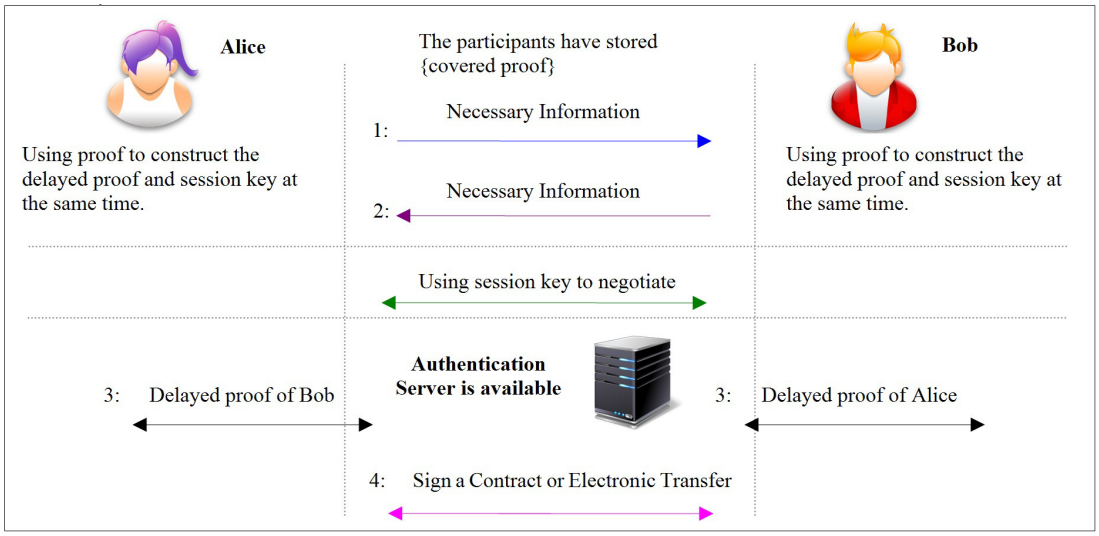


Figure 4: The framework of the delayed authentication

Step 3. Alice and Bob communicates with each other and both of them store the other's delayed proof securely.

The second phase commit with security services (Delayed Authentication).

Step 1. Alice sends $\{ID_A, m_B\}$ to the authentication server in order to authenticate Bob. The same way for Bob.

Step 2. Upon receiving $\{ID_A, m_B\}$ from Alice, the server will compute $H' = H(H(ID_B||K)||T_b(x)||ID_B)$ based on its secret key and the user's identity. Then the server will check if $H' = DP_B$? If holds, the server finishes the task of Bob's authentication. Finally, the server computes $V_A = H(H(ID_A||K)||T_a(x)||ID_S)$ and sends $\{ID_S, V_A\}$ to Alice. The same way for Bob.

Step 3. Upon receiving $\{ID_S, V_A\}$ from the server, Alice firstly checks V_A to authenticate the server. If the server is passed validation, Alice will confirm that Bob is the legal and real "Bob". The same way for Bob.

After the second phase is performed, Alice and Bob can continue to do the important things.

5 Security Consideration

5.1 Security Analysis for Local Authentication

The security of this kind local authentication scheme is based on one way secure hash function. The scheme aims at authenticate oneself efficiently with one-factor authentication. So, this scheme may be sacrifice some security for improving efficiency. For example, losing smart device and guessing attack(An adversary gets the user's smart device and then carries out the guessing attacks.) may deal with this scheme: An adversary gets the smart device and reads the information $\{V, L, R\}$. Then the adversary guesses a password PW^* to compare $H(H(PW^*||R) \oplus V)$ with L repeatedly until guessing the right password.

So, in order to resist losing smart device and guessing attack, we choose two-factor local authentication scheme based on biometric and password. This kind of scheme must be verified by two-factor authentication which can lead to some computations and hardware spending.

5.2 Security Analysis for Delayed Authentication

For simplicity, we only discuss the delayed authentication with password-based, and we do not design with biometric. In this section, there is no local proof stored in smart device in our scheme for resisting lost smart device and guessing attack. So, we only use covered proof to construct the delayed proof. From the **Table 2**, we can see that the proposed scheme can provide known secure session key agreement, impersonation attack and so on.

5.3 Security Proof Based on the BAN Logic [1] for Delayed Authentication

For convenience, we first give the description of some notations (Table 3) used in the BAN logic analysis and define some main logical postulates (Table 4) of BAN logic. We combine the two phases (Generate delayed proof with a session key, Delayed Authentication) together to prove, because only in the second phase the two involved users can just do the important things.

According to analytic procedures of BAN logic and the requirement of delayed authentication scheme, our CMFC

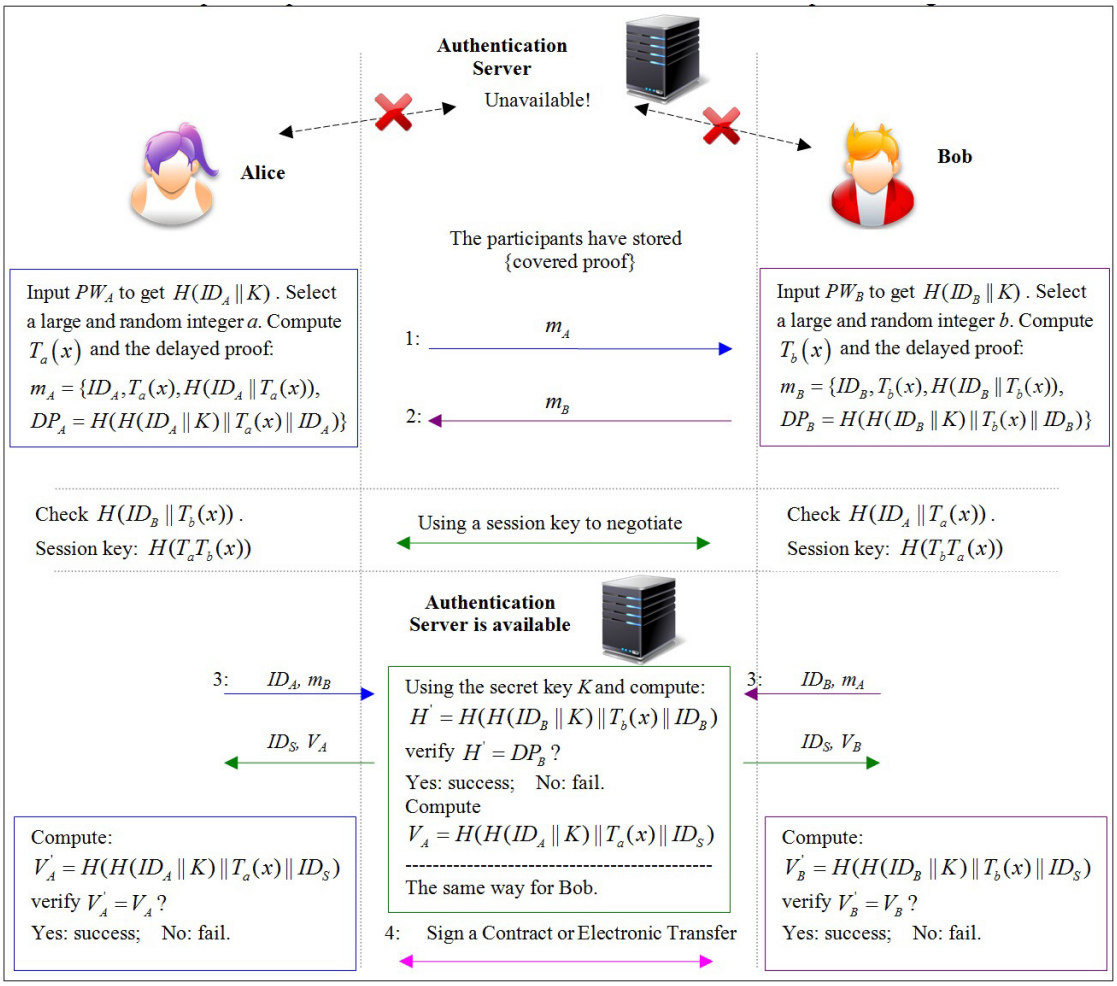


Figure 5: The delayed authentication with password-based and chaotic maps

Table 2: Definition and simplified proof with combining the two phases in section 4.2

Attack Type	Security Requirements	Definition	Simplified Proof	Hard Problems
Automatic validation attacks	Guessing attacks (On-line or off-line)	In an off-line guessing attack, an attacker guesses a password or long-term secret key and verifies his/her guess, but he/she does not need to participate in any communication during the guessing phase. In an undetectable on-line guessing attack, an attacker searches to verify a guessed password or long-term secret key in an on-line transaction and a failed guess cannot be detected and logged by the server.	There is no any match value to compare	Secure one way hash
	Losting smart device and guessing attacks	An adversary gets the users smart device and then carries out the guessing attacks.	There is no any match value to compare	Secure one way hash
No freshness verify attacks	Perfect forward secrecy	An authenticated key establishment protocol provides perfect forward secrecy if the compromise of both of the nodes secret keys cannot results in the compromise of previously established session keys.	Different session has different nonces.	Chaotic maps problems
	Known session key security	Each execution of the protocol should result in a unique secret session key. The compromise of one session key should not compromise the keys established in other sessions.	Different session has different nonces.	Chaotic maps problems
Missing encrypted identity attacks	Man-in-the-middle attack(MIMA)	The MIMA attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.	All the information includes the ID and some nonces: a, b and the another form $T_a(x), T_b(x)$.	Chaotic maps problems
	Impersonation attack	An adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications protocol.	All the information includes the ID and some nonces: a, b and the another form $T_a(x), T_b(x)$.	Chaotic maps problems
No freshness verify attacks	Replay attack	A replay attack is a form of network attack in which a valid data transmission is repeated or delayed maliciously or fraudulently.	Every important message includes the nonces: a, b and the another form $T_a(x), T_b(x)$.	Chaotic maps problems
Design defect attacks	Stolen-verifier attacks	An adversary gets the verifier table from servers by a hacking way, and then the adversary can launch any other attack which called stolen-verifier attacks.	There are no any verification tables in any node.	Natural Resistance

Table 3: Notations of the BAN logic

Symbol	Definition
$P\mid\equiv X$	The principal P believes a statement X, or P is entitled to believe X.
$\#(X)$	The formula X is fresh.
$P\mid\Rightarrow X$	The principal P has jurisdiction over the statement X.
$P\lhd X$	The principal P sees the statement X.
$P\mid\sim X$	The principal P once said the statement X.
(X,Y)	The formula X or Y is one part of the formula (X,Y).
$\langle X\rangle_Y$	The formula X combined with the formula Y.
X_K	The formula X is encrypted under the key K.
$(X)_K$	The formula X is hash function with the key K. If there is no K, and that means is no key input.
$P \overset{K}{\longleftrightarrow} Q$	The principals P and Q use the shared key K to communicate. The key K will never be discovered by any principal except P and Q.
$\overset{K}{\longrightarrow} P$	The public key of P, and the secret key is described by K^{-1}.

Table 4: Logical postulates of the BAN logic

Symbol	Definition
$\frac{P\mid\equiv P \overset{K}{\longleftrightarrow} Q, P\{X\}_K}{P\mid\equiv Q\mid\sim X}$	R1: The message-meaning rule
$\frac{P\mid\equiv\#(X)}{P\mid\equiv\#(X,Y)}$	R2: The freshness-conjuncatenation rule
$\frac{P\mid\equiv\#(X), P\mid\equiv Q\mid\sim X}{P\mid\equiv Q\mid\equiv X}$	R3: The nonce-verification rule
$\frac{P\mid\equiv Q\mid\Rightarrow X, P\mid\equiv Q\mid\equiv X}{P\mid\equiv X}$	R4: The jurisdiction rule
$\frac{P\mid\equiv Q\mid\equiv(X,Y)}{P\mid\equiv Q\mid\equiv X}$	R5: The belief rules
Molecule can deduce denominator for above formulas.	

scheme should satisfy the following goals in Table 5.

Table 5: Goals of the proposed scheme

Goal 1. $U_A\mid\equiv(U_A \overset{SK}{\longleftrightarrow} U_B)$;
Goal 2. $U_A\mid\equiv U_B\mid\equiv(U_A \overset{SK}{\longleftrightarrow} U_B)$;
Goal 3. $U_B\mid\equiv(U_A \overset{SK}{\longleftrightarrow} U_B)$;
Goal 4. $U_B\mid\equiv U_A\mid\equiv(U_A \overset{SK}{\longleftrightarrow} U_B)$;

First of all, we transform the process of our protocol to the following idealized form. Because only the second phase can make the two users (Alice and Bob) do the important, we just begin with the authentication server is available to analyse.

For Alice and authentication server:

$(U_A \rightarrow Server)m_1$: $Server \lhd ID_A$, ID_B, $T_b(x)$, $(ID_B||T_b(x))$, $((H(ID_B||K))||T_b(x)||ID_B)$;

$(Server \rightarrow U_A)m_2$: $U_A \lhd ID_S$, $(H(ID_A \mid\mid K) \mid\mid T_a(x) \mid\mid ID_S)$.

For Bob and authentication server:

$(U_B \rightarrow Server)m_3$: $Server \lhd ID_A$, ID_B, $T_a(x)$, $(ID_A||T_a(x))$, $((H(ID_A||K))||T_a(x)||ID_A)$;

$(Server \rightarrow U_B)m_4$: $U_B \lhd ID_S$, $(H(ID_B \mid\mid K) \mid\mid T_b(x) \mid\mid ID_S)$.

According to the description of our protocol, we could make the following assumptions about the initial state, which will be used in the analysis of our protocol in Table 6.

Based on the above assumptions, the idealized form of our scheme is analyzed as follows. We only analyze the process of Alice and authentication server, and the same way for Bob and authentication server.The main steps of the proof are described as follows:

For m_1:

According to m_1 and P_1, P_5 and relating with R_1, we could get: $S_1 : U_A \mid\equiv Server \mid\sim m_1$.

Based on m_1 and the initial assumptions P_1, P_3, P_4, P_5, P_7, we could get: $S_2 : Server \mid\equiv \#m_1$.

Combine $S_1, S_2, P_3, P_4, P_5, P_7, R_2$, we could get: S_3: $Server \mid\equiv \#\ ID_A$, ID_B, $T_b(x)$, $((H(ID_B \mid\mid K)) \mid\mid T_b(x) \mid\mid ID_B)$.

Based on R_3, we take apart S_3 and get: S_4 : $Server \mid\equiv \#T_b(x)$, $S_5 : Server \mid\equiv \#((H(ID_B||K)) \mid\mid T_b(x) \mid\mid ID_B)$.

Based on the secret key K and S_2, the server can authenticate Bob by computing the new hash value to compare with S_5. If the server authenticate Bob, it will continue to the process of m_2.

For m_2:

According to m_2 and P_1, P_7 and relating with R_1, we could get: $S_6 : Server \mid\equiv U_A \mid\sim m_2$.

Based on m_2 and the initial assumptions P_1, P_3, P_4, P_5, P_7, we could get: $S_7 : U_A \mid\equiv \#m_2$.

Table 6: Assumptions about the initial state of our protocol

Initial states	
$P_1 : U_A\| \equiv \overset{T_K(x)}{\longrightarrow} Server$	$P_2 : U_B\| \equiv \overset{T_K(x)}{\longrightarrow} Server$
$P_3 : U_A\| \equiv \#(a)$	$P_4 : U_B\| \equiv \#(b)$
$P_5 : U_A\| \equiv U_A \overset{H(ID_A\|\|K)}{\longleftrightarrow} Server$	$P_6 : U_B\| \equiv U_B \overset{H(ID_B\|\|K)}{\longleftrightarrow} Server$
$P_7 : Server\| \equiv U_A \overset{H(ID_A\|\|K)}{\longleftrightarrow} Server$	$P_8 : Server\| \equiv U_B \overset{H(ID_B\|\|K)}{\longleftrightarrow} Server$

Combine $S_6, S_7, P_3, P_4, P_5, P_7, R_2$, we could get: S_8 : $U_A \mid\equiv \#ID_S, (H(ID_A||K)||T_a(x)||ID_S)$.

Based on R_3, we take apart S_8 and get: S_9 : $Server \mid\equiv \#(H(ID_A||K)||T_a(x)||ID_S)$.

Based on P_5, P_7 and S_9, Alice can authenticate the server by computing the new hash value to compare with S_9.

Combine:

Because the Alice, Bob and the Server communicate each other just now, they confirm the other is on-line. And based on S_9, R_4 with chaotic maps problems, we could get:

Goal 1. $U_A\| \equiv (U_A \overset{SK}{\longleftrightarrow} U_B)$;

Goal 2. $U_A\| \equiv U_B\| \equiv (U_A \overset{SK}{\longleftrightarrow} U_B)$.

The same way for Bob and the server, we could get:

Goal 3. $U_B\| \equiv (U_A \overset{SK}{\longleftrightarrow} U_B)$;

Goal 4. $U_B\| \equiv U_A\| \equiv (U_A \overset{SK}{\longleftrightarrow} U_B)$.

According to (Goal 1 $\sim$ Goal 4), we know that both Alice and Bob believe that the Server can authenticate them and the session key is fresh based on the fresh nonces a, b.

6 Efficiency Analysis

6.1 The Comparisons Among Different Algorithms

Compared to RSA, ECC and Bilinear map, Chebyshev polynomial computation problem offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings. Chaotic maps encryption algorithm: As a special form of motion, Chaos means that in a certain nonlinear system can appear similar to the behavior of random phenomena without needing any random factors. Chaotic system has the characteristics of certainty, boundness, sensibility to initial parameters and unpredictability, etc. Chaotic maps encryption algorithm utilizes the unique semi-group mature of Chebyshev chaotic maps, based on two difficult problems-the chaotic maps discrete logarithm problem and the chaotic maps Diffie-Hellman problem, puts forward a kind of encryption algorithm. Compared with ECC encryption algorithm, Chaotic maps encryption algorithm avoids scalar multiplication and modular exponentiation computation, effectively improves the efficiency. However, Wang [13] proposed several methods to solve the Chebyshev polynomial computation problem. To be more precise, on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, the computational time of a one-way hashing operation, a symmetric encryption/decryption operation, an elliptic curve point multiplication operation and Chebyshev polynomial operation is 0.0005s, 0.0087s, 0.063075s and 0.02102s separately [7]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations. According to the results in [11], one pairing operation requires at least 10 times more multiplications in the underlying finite field than a point scalar multiplication in ECC does in the same finite field.

Through the above mentioned analysis, we can reached the conclusion approximately as follows:

$$T_p \approx 10T_m, T_m \approx 3T_c, T_c \approx 2.42T_s, T_s \approx 17.4T_h,$$

we sum up these formulas into one so that it can reflect the relationship among the time of algorithms intuitively.

$$T_p \approx 10T_m \approx 30T_c \approx 72.6T_s \approx 1263.24T_h,$$

where T_p: Time for bilinear pair operation, T_m: Time for a point scalar multiplication operation, T_c: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial, T_s: Time for symmetric encryption algorithm, T_h: Time for Hash operation.

About these algorithms, our proposed CMFC scheme only used the chaotic cipher and a secure one way hash as the main algorithm (see Table 7) which are more efficient bilinear pair operation and a point scalar multiplication operation ECC-based. Especially for hash operation, it can be ignored compared with the other three algorithms.

6.2 The Sum Up About Our Scheme's Efficiency

Because there are no any related literatures, we can not give any comparisons about efficiency. From **Table 7**, we can conclude that our CMFC scheme has high-efficient property.

Table 7: Our proposed scheme's efficiency

Type	Instance	Party	Ours CMFC scheme	
Local Proof	Only password	Server	Registration	$2T_h$
			Local authentication	No need
		Alice	Registration	$1T_h$
			Local authentication	$2T_h$
	Biometric and password	Server	Registration	$1T_h$
			Local authentication	No need
		Bob	Registration	$1T_h$
			Local authentication	$1T_h$+$1T_s$
Delayed Proof	Only password	Server	Unavailable Server Phase	No need
			Delayed authentication	$4T_h$
		Alice	Unavailable Server Phase	$5T_h$+$1T_c$
			Delayed authentication	$2T_h$
		Bob	Unavailable Server Phase	$5T_h$+$1T_c$
			Delayed authentication	$2T_h$

T_h:Time for Hash operation
T_s:Time for symmetric parametric function
T_c:The time for executing the $T_n(x)$ mod p in Chebyshev polynomial using the algorithm in literature [9].

7 Conclusions

In this paper, we propose CMFC, a novel idea towards resisting fragile communications which is divided into the framework and the instance. The framework is the macrostructure which can be implemented by many algorithms, such as RSA, ECC and Bilinear map. For clearing expression of the framework, we illustrate three instances: For local proof framework, we use one factor (password) and two-factor (biometric with password) to construct two instances for adapting to different environment. For delayed proof framework, we only use one factor (password) to construct a instance due to limited space. In addition, we give some new definitions about the meanings of different proofs. Security consideration and efficiency analysis are also focused on discussion. CMFC is not a panacea, but it offers reasonable security, preferable efficiency, easy usability, and appears to fit well with some practical applications – when the authentication server is unavailable. We often face the fragile communications, so we must do something, that is the core motivation of this paper.

Acknowledgement

This work is supported by the Liaoning Provincial Natural Science Foundation of China (Grant No. 201602680).

References

[1] P. Barreto, B. Lynn, M. Scott, "On the selection of pairing-friendly groups," in *Selected Areas in Cryptography*, LNCS 3006, pp.17–25, Springer-Verlag, 2004.

[2] A. Bhargav-Spantzel, A. C. Squicciarini, S. Modi, M. Young, E. Bertino and S. J. Elliott, "Privacy preserving multi-factor authentication with biometrics," *Journal of Computer Security*, vol. 15, no. 5, pp. 529–560, 2007.

[3] S. Bradner, "What a fragile communications web we've woven," *Network World*, vol. 25, no. 6, pp.33, 2008.

[4] D. Chhabra, S. Zhao, W. Lee and N. Okamoto, "Negotiated self-authenticated experience and homeland travel loyalty: implications for relationship marketing," *Anatolia*, vol. 23, no. 3, pp. 429–436, 2012.

[5] X. Huang, X. Yang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1390–1397, 2011.

[6] X. Huang, X. Yang, E. Bertino, J. Zhou and X. Li, "Robust Multi-Factor Authentication for Fragile Communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2014.

[7] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li and A. Alelaiwi, "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dynamics*, vol. 83, no. 4, pp. 2085-2101, Mar. 2016.

[8] L. Kocarev, and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*, Springer, 2011.

[9] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos Solitons Fractals*, vol. 37, no. 3, pp. 669–674, 2008.
[10] H. Wang, H. Zhang, J. Li and X. Chen, "A(3,3) visual cryptography scheme for authentication," *Journal of Shenyang Normal University (Natural Science Edition)*, vol. 31, no. 101(03), pp. 397-400, 2013.
[11] X. Wang, and J. Zhao, "An improved key agreement protocol based on chaos," *Communication Nonlinear Science Number Simulation*, vol. 15, pp. 4052–4057, 2010.
[12] G. Yang, D. S. Wong, H. Wang and X. Deng, "Two-factor mutual authentication based on smart cards and passwords," *Journal of Computer and System Sciences*, vol. 74, no. 7, pp. 1160–1172, 2008.
[13] H. Zhu, Y. Zhang, Y. Zhang and H. Li, "A novel and provable authenticated key agreement protocol with privacy protection based on chaotic maps towards mobile network," *International Journal of Network Security*, vol. 18, no. 1, pp. 116–123, Jan. 2016.
[14] H. Zhu, Y. Zhang, Y. Xia, and H. Li, "Password-authenticated key exchange scheme using chaotic maps towards a new architecture in standard model," *International Journal of Network Security*, vol. 18, no. 2, pp. 326–334, Mar. 2016.
[15] H. Zhu, Y. Zhang, H. Li, and L. Lin, "A novel biometrics-based one-time commitment authenticated key agreement scheme with privacy protection for mobile network," *International Journal of Network Security*, vol. 18, no. 2, pp. 209–216, Mar. 2016.

Yang Sun obtained his master degree in Information Science and Engineering from Northeastern University. Yang Sun is a full associate professor of the Kexin software college at Shenyang Normal University. He is also a department head of network engineering. He has research interests in wireless networks, mobile computing, cloud computing, social networks and network security. Yang Sun had published more than 15 international journal and conference papers on the above research fields.

Xueshuai Feng graduated with a Bachelor of Engineering from Shenyang Normal University in 2015. In his college, after completing the learning task, he interests in exploring his professional knowledge. During graduate, under the guidance of his master instructor, he researches IoT security theory and technology.

Hongfeng Zhu obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hongfeng Zhu is a full associate professor of the Kexin software college at Shenyang Normal University.He is also a master's supervisor. He has research interests in wireless networks, mobile computing, cloud computing,social networks, network security and quantum cryptography. Dr. Zhu had published more than 50 international journal papers (SCI or EI journals) on the above research fields.

Identity Based Proxy Signature from RSA without Pairings

Lunzhi Deng[1,2,3], Huawei Huang[1], and Yunyun Qu[1]
(Corresponding author: Lunzhi Deng)
School of Mathematics Science, Guizhou Normal University[1]
School of Big Data and Computer Science, Guizhou Normal University[2]
Guiyang 550001, China
Guizhou Provincial Key Laboratory of Public Big Data[3]
Guiyang 550025, China
(Email: denglunzhi@163.com)
(Received Sept. 16, 2015; revised and accepted Jan. 23 & Mar. 29, 2016)

Abstract

RSA is a key cryptography technique and provides various interfaces for the applied software in real-life scenarios. Although some good results were achieved in speeding up the computation of pairing function in recent years, the computation cost of the pairings is much higher than that of the exponentiation in a RSA group. So it is still interesting to design efficient cryptosystems based on RSA primitive. A proxy signature scheme allows a proxy signer to sign messages on behalf of an original signer within a given context. Most identity based proxy signature schemes currently known employ bilinear pairings. In this paper, an identity based proxy ring signature (IBPS) scheme from RSA without pairings is constructed, and the security is proved under the random oracle model.
Keywords: Identity based cryptography, proxy signature, RSA

1 Introduction

Public key cryptography is an important technique to realize network and information security. In traditional public key infrastructure, each user generates his own secret and public keys [13]. A certification authority must sign a digital certificate which links the identity of the user and his public key. The validity of this certificate must be checked before using the public key of the user, when encrypting a message or verifying a signature. Obviously, the management of digital certificates decreases the efficiency of practical implementations of public key cryptosystem. To solve the problem, Shamir [21] defined a new public key paradigm called identity-based public key cryptography. In this system, each user needs to register at a trusted private key generator (PKG) with identity of himself before joining the network. Once a user is accepted, the PKG will generate a private key for the user and the user's identity (e.g., user's name or email address) becomes the corresponding public key. In order to verify a digital signature or send an encrypted message, a user needs to only know the identity of communication partner and the public key of the PKG [20].

Shamir [21] proposed an identity-based signature scheme from the RSA primitive. Guillou and Quisquater [6] proposed a similar RSA identity-based signature scheme, which is constructed from a zero-knowledge identification protocol. Herranz [8] proposed an identity-based ring signatures from RSA whose security is based on the hardness of the RSA problem. After initial schemes, the following breakthrough result in the area of identity-based cryptography came in 2001, when Boneh and Franklin [2] designed an efficient identity-based public key encryption scheme. In the design, they used as a tool bilinear pairings, a kind of maps which can be constructed on some elliptic curves. Using bilinear pairings, a lot of identity-based schemes have been proposed for encryption, signature, key agreement, etc. However, it is still desirable to find schemes for identity-based scenarios which do not need to employ bilinear pairings.

The concept of proxy signatures was first introduced by Mambo et al. [16]. Based on the delegation type, proxy signature schemes are classified into three types: full delegation, partial delegation and delegation by warrant. In a full delegation scheme, the original signer's private key is given to the proxy signer. Hence the proxy signer has the same signing right as the original signer. Obviously, such schemes are impractical and insecure for most of real-world settings. In a partial delegation scheme, a proxy signer has a new key, called proxy private key, which is different from the original signer's private key. Although proxy signatures generated by using proxy private key are different from the original signer's standard signatures,

the proxy signer is not limited on the range of messages he can sign. This weakness is eliminated in delegation by warrant schemes. One of the main advantages of the use of warrants is that it is possible to include any type of security policy (that specifies what kinds of messages are delegated, and may contain other information, such as the identities of the original signer and the proxy signer, the delegation period, etc.) in the warrant to describe the restrictions under which the delegation is valid. Therefore, proxy signature schemes which use the method of this approach attract a great interest, and it is often expected that new proxy signature schemes will implement the functionality of warrants.

In order to adapt different situations, many proxy signature variants are produced, such as one-time proxy signature, proxy blind signature, multi-proxy signature, and so on. Using bilinear pairings, people proposed many new ID-based proxy signature (IBPS) scheme [1, 4, 5, 9, 10, 11, 14, 15, 17, 18, 19, 24, 25, 26]. All the above IBPS schemes are very practical, but they are based on bilinear pairings and the pairing is regarded as the most expensive cryptography primitive. Recently, He et al. [7] proposed an ID-based proxy signature schemes without bilinear pairings. Tiwari and Padhye [22] proposed a provable secure multi-proxy signature scheme without bilinear maps. Kim et al. [12] constructed a provably secure ID-based proxy signature scheme based on the lattice problems. The computation cost of the pairings is much higher than that of the exponentiation in a RSA group. Therefore, IBPS schemes based on RSA primitive would still be appealing.

2 Preliminaries

The notations of this paper are listed in the following:

- N: A large composite number, the product of two prime numbers p, q.
- b: A prime number satisfying $gcd(b, \varphi(N)) = 1$.
- p, q, a: The master key satisfying $N = pq$ and $a = b^{-1} \bmod \varphi(N)$.
- ID_i/D_i: The user's identity /private key.
- ID_o/D_o: The original signer's identity/private key.
- ID_p/D_p: The proxy signer's identity/private key.
- m_w: The warrant consisting of the identities of original signer and proxy signer, the delegation duration and so on.
- π: The proxy delegation.
- H_0, H_1, H_2: Three hash functions.

We define the RSA problem as follows.

Definition 1. *Let $N = pq$, where p and q are two k-bit prime numbers. Let b be a random prime number, greater than 2^l for some fixed parameter l, such that $\gcd(b, \varphi(N)) = 1$. Given $Y \in Z_N^*$, RSA problem is to find $X \in Z_N^*$ such that $X^b = Y \bmod N$.*

An identity based proxy signature scheme consists of the following six algorithms:

- Setup: This algorithm takes as input a security parameter k, then returns *params* (system parameters) and a randomly chosen master secret key *msk*. After the algorithm is performed, the PKG publishes the system parameters *params* and keeps the master key *msk* secret.
- Key extract: This algorithm takes as input *params*, *msk*, identity $ID_i \in \{0,1\}^*$ of an entity, and returns a private key D_i. The PKG carries out the algorithm to generate the private key D_i and send D_i to the corresponding owner ID_i via a secure channel.
- Delegate: This algorithm takes as input the *params*, original signer's private key D_o, a warrant m_w and outputs the delegation π.
- Delegation verify: This algorithm takes as input *params*, π and verifies whether π is a valid delegation from the original signer.
- Proxy sign: This algorithm takes as input the *params*, proxy signer's private key D_p, delegation π, a message m and outputs the proxy signature σ.
- Proxy signature verify: This algorithm takes as input the original signer's identity ID_o, the proxy signer's identity ID_p, a proxy signature σ, and outputs 1 if the proxy signature is valid or 0 otherwise.

Definition 2. *An identity based proxy signature scheme is unforgeable(UNF-IBPS) if no polynomially bounded adversary has a non-negligible advantage in the following game.*

Game. *Now we illustrate the game performed between a challenger $\mathscr{C}$ and a adversary $\mathscr{A}$ for an identity based proxy signature scheme.*

Initialization. *$\mathscr{C}$ runs the setup algorithm to generate a master secret key msk and the public system parameters params. $\mathscr{C}$ keeps msk secret and gives params to $\mathscr{A}$.*

Query. *$\mathscr{A}$ performs a polynomially bounded number of queries. These queries may be made adaptively, i.e. each query may depend on the answers to the previous queries.*

- *Hash functions query: $\mathscr{A}$ can ask for the values of the hash functions for any input.*
- *Key query: When $\mathscr{A}$ requests the private key of the user ID_i, $\mathscr{C}$ responds with the private key D_i.*

- *Delegation query: When $\mathscr{A}$ submits original signer's identity ID_o and a warrant m_w to the challenger, $\mathscr{C}$ responds by running the delegate algorithm on the warrant m_w, the original signer's private key D_o.*
- *Proxy signature query: When $\mathscr{A}$ submits original signer's identity ID_o, proxy signer's identity ID_p, a warrant m_w and a message m to the challenger, $\mathscr{C}$ responds by running the proxy sign algorithm on the message m, the warrantm_w, the private keys D_o and D_p.*

Forge. *$\mathscr{A}$ outputs a tuple (π^*, ID_o) or $(m^*, m_w^*, \sigma^*, ID_o, ID_p)$. $\mathscr{A}$ wins the game, if one of the following cases is satisfied:*

Case 1: *The final output is (π^*, ID_o) and it satisfies:*

1) *π^* is a valid delegation.*
2) *$\mathscr{A}$ does not query the original signer ID_o's private key.*
3) *π^* is not generated from the delegation query.*

Case 2: *The final output is $(m^*, m_w^*, \sigma^*, ID_o, ID_p)$ and it satisfies:*

1) *σ^* is a valid proxy signature.*
2) *$\mathscr{A}$ does not query the original signer ID_o's private key.*
3) *The tuple (ID_o, ID_p, m_w^*) is not appear in delegation query.*
4) *σ^* is not generated from the proxy signature query.*

Case 3: *The final output is $(m^*, m_w^*, \sigma^*, ID_o, ID_p)$ and it satisfies:*

1) *σ^* is a valid proxy signature.*
2) *$\mathscr{A}$ does not query the proxy signer ID_p's private key.*
3) *σ^* is not generated from the proxy signature query.*

The success probability of $\mathscr{A}$ is defined as: $Succ_{\mathscr{A}}^{UNF-IBPS} = Pr[\mathscr{A}\ win]$.

3 The Proposed Scheme

Setup: Given security parameters k, a trusted private key generator (PKG) generates two random k-bit prime numbers p and q. Then it computes $N = pq$. For some fixed parameter l (for example $l = 200$), chooses at random a prime number b satisfying $2^l < b < 2^{l+1}$ and $\gcd(b, \varphi(N)) = 1$, and computes $a = b^{-1} \bmod \varphi(N)$. Furthermore, PKG chooses cryptographic hash functions described as follows: $H_0 : \{0,1\}^* \to Z_N^*$, $H_1 : \{0,1\}^* \times Z_N^* \to Z_b^*$ and $H_2 : \{0,1\}^* \times Z_N^* \times Z_N^* \to Z_b^*$. The set of public parameters is: $params = \{N, b, H_0, H_1, H_2\}$, the secret information stored by PKG is master-key=(p, q, a).

Key extract: For an identity $ID_i \in \{0,1\}^*$, PKG computes $D_i = Q_i^a, Q_i = H_0(ID_i)$ and sends D_i to the user ID_i via a secure channel.

Delegate: m_w is the warrant consisting of the identities of original signer and proxy signer, the delegation duration and so on. On input the warrant m_w, the original signer ID_o, whose private key is D_o, performs the following steps:

1) Randomly selects $A \in Z_N^*$, computes $T = A^b \bmod N$, $h = H_1(m_w, T)$.
2) Computes $R = AD_o^h \bmod N$.
3) Outputs $\pi = (m_w, T, R)$ as the delegation.

Delegation verify: To verify a delegation $\pi = (m_w, T, R)$ for an identity ID_o, the verifier performs the following steps:

1) Computes $h = H_1(m_w, T)$.
2) Checking whether $R^b = TQ_o^h \bmod N$. If the equality holds, accepts the delegation. Otherwise, rejects.

Proxy sign: For a message m, the proxy signer (whose identity is ID_p) who owns the delegation $\pi = (m_w, T, R)$ does the following:

1) Randomly selects $B \in Z_N^*$, computes $S = B^b \bmod N$, $k = H_2(m, m_w, T, S)$.
2) Computes $Z = RBD_p^k \bmod N$.
3) Outputs the signature $\sigma = (m, m_w, T, S, Z)$.

Proxy signature verify: To verify the validity of a proxy signature (where the original singer's identity is ID_o, the proxy singer's identity is ID_p), a verifier first checks whether the original signer and proxy signer conform to m_w, then performs the following steps.

1) Computes $h = H_1(m_w, T)$ and $k = H_2(m, m_w, T, S)$.
2) Checking whether $Z^b = TSQ_o^h Q_p^k \bmod N$. If the equality holds, outputs 1. Otherwise, outputs 0.

On correctness, we have $Z^b = (RBD_p^k)^b = R^b B^b D_p^{bk} = TSQ_o^h Q_p^k$.

4 Security of The Proposed Scheme

Theorem 1. *The scheme is unforgeable against the adversary in randomly oracle model if the RSA problem is hard.*

Proof. Suppose the challenger $\mathscr{C}$ receives a random instance (N, b, Y) of the RSA problem and and has to find an element $X \in Z_N^*$ such that $X^b = Y$. $\mathscr{C}$ will run $\mathscr{A}$ as a subroutine and act as $\mathscr{A}$'s challenger in the UNF-IBPS game.

Initialization. At the beginning of the game, $\mathscr{C}$ runs the setup program with the parameter k, gives $\mathscr{A}$ the system parameters params=$\{N, b, H_0, H_1, H_2\}$.

Queries. Without loss of generality, it is assumed that all the queries are distinct and $\mathscr{A}$ will ask for $H_0(ID)$ before ID is used in any other queries. $\mathscr{C}$ will set several lists to store the queries and answers, all of the lists are initially empty.

- H_0 queries: $\mathscr{C}$ maintains the list L_0 of tuple (ID_i, A_i). When $\mathscr{A}$ makes a query $H_0(ID_i)$, $\mathscr{C}$ responds as follows.
  At the $j^{th} H_0$ query, $\mathscr{C}$ set $H_0(ID^*) = Y$. For $i \neq j$, $\mathscr{C}$ randomly picks a value $A_i \in Z_N^*$ and sets $H_0(ID_i) = A_i^b$, then the query and the answer will be stored in the list L_0.
- H_1 queries: $\mathscr{C}$ maintains the list L_1 of tuple (α_i, h_i). When $\mathscr{A}$ makes a query $H_1(\alpha_i)$. $\mathscr{C}$ randomly picks a value $h_i \in Z_b^*$ and sets $H_1(\alpha_i) = h_i$, then the query and the answer will be stored in the list L_1.
- H_2 queries: $\mathscr{C}$ maintains the list L_2 of tuple (β_i, k_i). When $\mathscr{A}$ makes a query $H_2(\beta_i)$. $\mathscr{C}$ randomly picks a value $k_i \in Z_b^*$ and sets $H_2(\beta_i) = k_i$, then the query and the answer will be stored in the list L_2.
- Key extraction queries: $\mathscr{C}$ maintains the list L_K of tuple (ID_i, D_i). When $\mathscr{A}$ makes private key extraction query for identity ID_i. If $ID_i = ID^*$, $\mathscr{C}$ fails and stops. Otherwise $\mathscr{C}$ finds the tuple (ID_i, A_i) in list L_0 and returns $D_i = A_i$ to $\mathscr{A}$.
- Delegate queries: When $\mathscr{A}$ submits ID_o, ID_p and m_w to challenger. $\mathscr{C}$ outputs a delegation as follows.
  If $ID_o \neq ID^*$, $\mathscr{C}$ gives a delegation by calling the delegate algorithm. Otherwise, $\mathscr{C}$ does as follows.
  1) Randomly selects $A \in Z_N^*$ and $h \in Z_b^*$.
  2) Computes $T = A^b Q_o^{-h}$ and $R = A$.
  3) Stores the relation $h = H_1(m_w, T)$. If collision occurs, repeats Steps (1)-(3).
  4) Outputs $\pi = (m_w, T, R)$ as the delegation.
- Proxy signature queries. When $\mathscr{A}$ submits a delegation $\pi = (m_w, T, R)$ message m to the challenger. $\mathscr{C}$ outputs an identity based proxy signature as follows.
  If $ID_p \neq ID^*$, $\mathscr{C}$ gives a signature by calling the proxy sign algorithm. Otherwise, $\mathscr{C}$ does as follow.
  1) Randomly selects $B \in Z_N^*$ and $k \in Z_b^*$.
  2) Computes $S = B^b Q_p^{-k}$ and $Z = RB$.
  3) Stores the relation $k = H_2(m, m_w, T, S)$. If collision occurs, repeats Steps (1)-(3).
  4) Outputs the proxy signature $\sigma = (m, m_w, T, S, Z)$.

Forge. $\mathscr{A}$ outputs a tuple $\{\pi^* = (m_w, T, R), ID_o\}$ or $\{\sigma^* = (m, m_w, T, S, Z), ID_o, ID_p\}$. There are three cases to consider:

Case 1. The final output is $\{\pi^* = (m_w, T, R), ID_o\}$ and the output satisfies the requirement of Case 1 as defined in UNF-IBPS game.

Solve RSA problem. In fact, π^* is the signature on m_w by ID_o. By the forking lemma for generic signature scheme [3], two delegations can be generated: (m_w, T, R) and (m_w, T, R'). Where $h = H_1(m_w, T)$, $h' = H_1'(m_w, T)$. If $ID_o = ID^*$, RSA problem can be solved as follow: The relation becomes $(R'R^{-1})^b = Y^{h'-h} \bmod N$. Since $h, h' \in Z_b$, then $|h' - h| < b$. By the element b is a prime number, so it holds $\gcd(b, h' - h) = 1$. This means that there exist two integers c and d such that $cb + d(h' - h) = 1$. Finally, the value $X = (R'R^{-1})^d Y^c \bmod N$ is the solution of the given instance of the RSA problem. In effect, $X^b = (R'R^{-1})^{bd} Y^{bc} = Y^{d(h'-h)} Y^{bc} = Y^{cb+d(h'-h)} = Y$.

Probability. Let $q_{H_i} (i = 0, 1, 2)$, q_K, q_D and q_S be the number of $H_i (i = 0, 1, 2)$ queries, private key queries, delegating queries and proxy signing queries, respectively.
The probability that $\mathscr{C}$ does not fail during the queries is $\frac{q_{H_0} - q_K}{q_{H_0}}$. The probability that $ID_o = ID^*$ is $\frac{1}{q_{H_0} - q_K}$. So the combined probability is $\frac{q_{H_0} - q_K}{q_{H_0}} \cdot \frac{1}{q_{H_0} - q_K} = \frac{1}{q_{H_0}}$. Therefore, the probability of $\mathscr{C}$ to solve the RSA problem is: $\frac{\epsilon}{q_{H_0}}$.

Case 2. The final output is $\{\sigma^* = (m, m_w, T, S, Z), ID_o, ID_p\}$ and the output satisfies the requirement of Case 2 as defined in UNF-IBPS game.

Solve RSA problem. By the forking lemma for generic signature scheme [3], two proxy signatures can be generated: (m, m_w, T, S, Z) and (m, m_w, T, S, Z'). Where $k = k' = H_2(m, m_w, T, S)$, $h = H_1(m_w, T)$, $h' = H_1'(m_w, T)$, and $h \neq h'$. If $ID_o = ID^*$, RSA problem can be solved as follow: The relation becomes $(Z'Z^{-1})^b = Y^{h'-h} \bmod N$. Since $h, h' \in Z_b$, the that $|h' - h| < b$. By the element b is a prime number, so it

holds $\gcd(b, h' - h) = 1$. This means that there exist two integers c and d such that $cb + d(h' - h) = 1$. Finally, the value $X = (Z'Z^{-1})^d Y^c \bmod N$ is the solution of the given instance of the RSA problem. In effect, $X^b = (Z'Z^{-1})^{bd} Y^{bc} = Y^{d(h'-h)} Y^{bc} = Y^{cb+d(h'-h)} = Y$.

Probability. Probability of success is same as the probability in Case 1.

Case 3. The final output is $\{\sigma^* = (m, m_w, T, S, Z), ID_o, ID_p\}$ and the output satisfies the requirement of Case 3 as defined in UNF-IBPS game.

Solve RSA problem. By the forking lemma for generic signature scheme [3], two proxy signatures can be generated: (m, m_w, T, S, Z) and (m, m_w, T, S, Z'). Where $h = h' = H_1(m_w, T)$, $k = H_2(m, m_w, T, S)$, $k' = H_2'(m, m_w, T, S)$, and $k \neq k'$. If $ID_p = ID^*$, RSA problem can be solved as follow: The relation becomes $(Z'Z^{-1})^b = Y^{k'-k} \bmod N$. Since $k, k' \in Z_b$, then $|k' - k| < b$. By the element b is a prime number, so it holds $\gcd(b, k' - k) = 1$. This means that there exist two integers c and d such that $cb + d(k' - k) = 1$. Finally, the value $X = (Z'Z^{-1})^d Y^c \bmod N$ is the solution of the given instance of the RSA problem. In effect, $X^b = (Z'Z^{-1})^{bd} Y^{bc} = Y^{d(k'-k)} Y^{bc} = Y^{cb+d(k'-k)} = Y$.

Probability. Probability of success is same as the probability in Case 1.

□

5 Efficiency and Comparison

In this section, we compare the performance of our scheme with several other schemes. some notations are defined as follows:

P: a pairing operation.

E_M: a modular exponentiation.

M_P: a pairing-based scalar multiplication.

M_E: an ECC-based scalar multiplication.

Through PIV 3-GHZ processor with 512-MB memory and a Windows XP operation system. He et al. [7] obtained the running time for cryptographic operations. To achieve 1024-bit RSA level security, they use the Tate pairing defined over a supersingular curve $E/F_p : y^2 = x^3 + x$ with embedding degree 2, where q is a 160-bit Solinas prime $q = 2^{159} + 2^{17} + 1$ and p is a 512-bit prime satisfying $p + 1 = 12qr$. To achieve the same security level, they employed the parameter secp160r1 [23], where $p = 2^{160} - 2^{31} - 1$. The running times are listed in Table 1.

Table 1: Cryptographic operation time (in milliseconds)

P	E_M	M_P	M_E
20.04	5.31	6.38	2.21

A simple method is used to evaluate the computation efficiency. For example, Wu et al.'s [25] scheme requires 6 pairing-based scalar multiplication operations and 8 pairing operations. So the resulting computation time is $6.38 \times 6 + 20.04 \times 8 = 198.60$.

Based on the above parameter and ways, the detailed comparison results of several different IBPS schemes are illustrated in Table 2.

6 Conclusion

A proxy signature scheme allows a proxy signer to sign messages on behalf of an original signer within a given context. Most IBPS schemes currently known employ bilinear pairings. RSA is a key cryptography technique and provides various interfaces for the applied software in real-life scenarios. Although some good results were achieved in speeding up the computation of pairing function in recent years, the computation cost of the pairings is much higher than that of the exponentiation in a RSA group. In this paper, an IBPS scheme from RSA was proposed and the security was proved in the random oracle model. The scheme needs not pairings and it is more efficient than previous ones using bilinear pairings. Due to the good properties of our scheme, it should be useful for practical applications.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grants 61562012, 11261060, 61462016. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature scheme for delegation of signing rights," *IACR ePrint Archive*, 2003. (`http://eprint.iacr.org/2003/096/`)

[2] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[3] M. Bellare and G. Neven, "Multisignatures in the plain publickey model and a general forking lemma," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06)*, pp. 390–399, 2006.

Table 2: Comparison of several IBPS schemes

Scheme	Delegate	D-Verify	Proxy signing	P-Verify	Time
Wu et al. [25]	$2M_P$	$3P$	$4M_P$	$5P$	198.60
Gu et al. [5]	$4M_P$	$4M_P + P$	$2M_P$	$4M_P + P$	129.40
Ji et al. [11]	$3M_P$	$2P$	$3M_P$	$M_P + 2P$	124.82
He et al. [7]	M_E	$3M_E$	M_E	$6M_E$	24.31
Our scheme	$2E_M$	$2E_M$	$2E_M$	$2E_M$	42.48

[4] C. Gu and Y. Zhu, "Provable security of ID-based proxy signature schemes," in *Networking and Mobile Computing*, LNCS 3619, pp. 1277–1286, Springer, 2005.

[5] C. Gu and Y. Zhu, "An efficient ID-based proxy signature scheme from pairings," in *Proceedings of Inscrypt'07*, LNCS 4990, Springer, pp. 40–50, 2007.

[6] L. C. Guillou and J. J. Quisquater, "A paradoxical identity-based signature scheme resulting from zero-knowledge," in *Proceedings of Crypto'88*, LNCS 403, pp. 216–231, 1988.

[7] D. B. He, J. H. Chen, and J. Hu, "An ID-based proxy signature schemes without bilinear pairings," *Annual Telecommunications*, vol. 66, pp. 657–662, 2011.

[8] J. Herranz, "Identity-based ring signatures from RSA," *Theoretical Computer Science*, vol. 389, pp. 100–117, 2007.

[9] M. S. Hwang, E. J. L. Lu, and I. C. Lin, "A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem," *IEEE Transactions On Knowledge and Data Engineering*, vol. 15, no. 6, pp. 1552–1660, 2003.

[10] X. Hu, W. Tan, H. Xu, and J. Wang, "Short and provably secure designated verifier proxy signature scheme," *IET Information Security*, vol. 10, no. 2, pp. 69–79, 2013.

[11] H. Ji, W. Han, and L. Zhao, "An identity-based proxy signature from bilinear pairings," in *WASE International Conference on Information Engineering*, pp. 14–17, 2009.

[12] K. S. Kim, D. Hong, and I. R. Jeong, "Identity-based proxy signature from lattices," *Journal Of Communications and Networks*, vol. 15, no. 1, pp. 1–7, 2013.

[13] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[14] B. Lee, H. Kim, and K. Kim, "Strong proxy signature and its applications," in *Proceedings of SCIS'01*, pp. 603–608, 2001.

[15] J. Y. Lee, J. H. Cheon, and S. Kim, "An analysis of proxy signatures: Is a secure channel necessary?" in *Proceedings of CT-RSA'03*, LNCS 2612, pp. 68–79, Springer, 2003.

[16] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: Delegation of the power to sign messages," *IEICE Transactions on Fundamentals*, vol. E79-A, no. 9, pp. 1338–1353, 1996.

[17] T. Malkin, S. Obana, and M. Yung, "The hierarchy of key evolving signatures and a characterization of proxy signatures," in *Proceedings of EUROCRYPT'04*, LNCS 3027, pp. 306–322, Springer, 2004.

[18] S. Mashhadi, "A novel non-repudiable threshold proxy signature scheme with known signers," *International Journal of Network Security*, vol. 15, no. 4, pp. 274–279, 2013.

[19] T. Okamoto, A. Inomata, and E. Okamoto, "A proposal of short proxy signature using pairing," in *International Conference on Information Technology (ITCC'05)*, pp. 631–635, 2005.

[20] K. R. Santosh, C. Narasimham, and P. Shetty, "Cryptanalysis of multi-prime RSA with two decryption exponents," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 40–44, 2016.

[21] A. Shamir, "Identity-based cryptosystem and signature scheme," in *Advances in Cryptology (Crypto'84)*, LNCS 196, pp. 47–53, Springer, 1984.

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

[23] The Certicom Corporation, *SEC2: Recommended Elliptic Curve Domain Parameters*, 2016. (`www.secg.org/collateral/sec2-final.pdf`)

[24] G. Wang, F. Bao, J. Zhou, and R. H. Deng, " Security analysis of some proxy signatures," in *Proceedings of ICISC'03*, LNCS 2971, pp. 305–319, Springer, 2003.

[25] W. Wu, Y. Mu, W. Susilo, J. Seberry, and X. Y. Huang, "Identity-based proxy signature from pairings," in *Proceedings of ATC'07*, LNCS 4610, pp. 22–31, Springer, 2007.

[26] J. Xu, Z. Zhang, and D. Feng, "ID-based proxy signature using bilinear pairings," in *PWorkshops on Parallel and Distributed Processing and Applications (ISPA'05)*, LNCS 3759, pp. 359–367, Springer, 2005.

Lunzhi Deng received his B.S. from Guizhou Normal University, Guiyang, China, in 2002; M.S. from Guizhou Normal University, Guiyang, China, in 2008; and Ph.D. from Xiamen University, Xiamen, China, in 2012. He is currently a Professor in the School of Mathematics

Science, Guizhou Normal University, Guiyang, China. His recent research interests include algebra and cryptography.

Huawei Huang received his B.S. from Jiangxi Normal University, Nanchang, China, in 2001; M.S. from Jiangxi Normal University, Nanchang, China, in 2004; and Ph.D. from Xidian University, Xi'an, China, in 2008. He is currently an Associate Professor in the School of Mathematics Science, Guizhou Normal University, Guiyang, China. His recent research interests include algebra and cryptography.

Yunyun Qu received his B.S. from Wuhan University, Wuhan, China, in 2005; M.S. from Southwest University, Chongqing, China, in 2009. He is currently an Associate Professor in the School of Mathematics Science, Guizhou Normal University, Guiyang, China. His recent research interests include number theory and cryptography.

Access Control Based Resource Allocation in Cloud Computing Environment

Junshe Wang[1], Jinliang Liu[2], and Hongbin Zhang[1]
(Corresponding author: Jinliang Liu)
School of Information Science and Engineering, Hebei University of Science and Technology[1]
26 Yuxiang Street, Yuhua District, Shijiazhuang City, Hebei Province, China
Communication System and Networks Department[2]
The 54th Research Institute of China Electronics Technology Group Corporation
589 Zhongshan West Street, Qiaoxi District, Shijiazhuang City, Hebei Province, China
(Email: 836251714@qq.com)
(Received Jan. 13, 2016; revised and accepted Apr. 17 & May. 31, 2016)

Abstract

In this paper, we propose a new dynamic resource allocation scheme - Access Control-based Resource Allocation (ACRA) for cloud users in order to address some deficiencies of the current resources allocation mechanisms in some free cloud computing environment. The proposed scheme comprehensively analyses behavior characteristics of cloud users and evaluates user behavior trust using fuzzy analytic hierarchy process (FAHP), and then dynamically adjusts the resources permission of cloud users according to their behavior trust values, thus effectively controlling user resource utilization. Experimental results show that the ACRA scheme can provide basis for allocating resources dynamically and reasonably to cloud users with different behaviors and improve resource utilization in cloud computing systems.

Keywords: Access control, cloud computing, dynamic permission adjustment, resource allocation

1 Introduction

Cloud computing is a new network computing model [16] based on virtualization technology and pay-on-demand business model, which can convert various types of resources (including hardware, platform and software) into services that can be used by cloud users with some special features, such as flexible expansion, dynamic resources allocation, and resource sharing. Dynamic resource allocation and sharing in a cloud computing environment is the common fundamental technology of IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service) [5, 7], thus forms a major direction of research on cloud computing technologies.

As an effective mechanism for controlling resource access in systems, access control can restrict resources utilization of access subjects according to their identities and predefined policies, thus effectively ensuring confidentiality, integrity, availability and legal of resources. It is one of the key mechanisms for isolating access behaviors of different users in a system in order to limit security risks. Therefore, access control can be used as a technical solution to the problems of resource allocation in cloud computing [4].

The existing access control technologies are mostly static authorization [2, 18], that is, after the subject receives access permission from the object, the permission can be used without restriction. However, a cloud computing environment, which is different from regular networking environment due to its dynamic and openness features, brings in new challenges to access control for resource allocation.

2 Related Work

This article has researched dynamic allocation and sharing strategies of cloud resources in-depth from the perspective of theory and practice.The authors of reference [6, 11, 22] take proactive measures to the long-term, predictable periodical loads, use statistic and machine learning methods to analyze statistical data about load changes and system logs and built a performance model under long-term load patterns. Although the model provides decision support for global multi-objective optimum making on resources, evaluating the capacity of resources by the average working time required to complete each task exists biggish uncertain. The author of reference [14] starts a research from the angle of resources reliability, considers the failure regularity characteristics of resources in time and space, and uses it as the basis for resources allocation, which shields a lot of fault resource nodes. But there are limits to this strategy. It does not deal with the

problem of resource expenditure minimization. In other words, it reduces the availability of resources when a resourceful node assigned to a task that requires very few resources. The author of reference [23] proposes a behavior-based resource provision policy for cloud computing. The policy can forecast the set of submitted task and expectation completing time of task at next time segment from the statistic results, and dynamically adjust the resource provision policy according to the policy table. This resource scheduling scheme does not give detailed analysis of user behavior characteristics.

At present, research focuses on the impact of load variation, the physical locations of resources or other factors on dynamic allocation and sharing of cloud resources, rarely involves the source of cloud computing demand and neglects the otherness among cloud users [10]. And in fact, in the process of cloud resource allocation, cloud user behaviors have a direct and significant impact on resource allocation, especially in some free cloud environments where cloud computing resources are only used by internal staff for free [20]. In such environments whether cloud resources are in an idle state becomes an important factor in judging the reasonability of resource utilization. If private cloud users do not undertake related tasks after acquiring cloud resources, it means they have seized these resources, and will affect resource utilization of other cloud users, as well as the resource utilization of the whole cloud computing system. In addition, malicious cloud computing users may utilize cloud resources perform malicious actions [21]. The behavior of some cloud users may change significantly over a period of time. All these factors will make well-behaved cloud users cannot get relevant cloud resources in time, which in turn affect the overall resource utilization of the cloud computing system [15].

The research presented in this paper aims at addressing the problem of degraded resource utilization in a private cloud environment caused by the equal permission granted to all cloud users. We propose a new access control-based scheme for cloud resource allocation that can not only improve the overall utilization of cloud resource, but also enhance security [9] for the cloud computing environment.

3 The ACRA Scheme Design

Inspired by the Role-based Access Control (RBAC) model reported in [1], in this paper we propose a new scheme of cloud resource allocation - Access Control-based Resource Allocation (ACRA). The overall framework of ACRA is shown in Figure 1.

The ACRA scheme consists of three key aspects. First, acquisition of cloud user behavior trust is a key component of the scheme. Both behavior of different users at the same moment and behavior of the same user under different circumstances are prodigiously different. Therefore, behavior trust can reflect behavior of cloud users; and accuracy of behavior trust evaluation directly impacts the subsequent authorization of cloud users [13]. The second key aspect of the scheme lies in, rules in the authorization process. Because the scheme is based on RBAC to solve problems in resource allocation in a cloud computing environment, the authorization rules involved must meet the requirements of the new environment. The third aspect of the ACRA scheme is to handle cloud user authorization required by the various behavior of cloud users in the cloud computing system. In this paper, we propose a dynamic permission adjustment mechanism on the basis of cloud users' behavior trust in order to manage their permission more flexibly, elastically and meticulously.

3.1 Behavior Trust Evaluation

3.1.1 Classification of Behavior Characteristics

The accuracy of behavior classification in a cloud computing environment has a direct impact on that of cloud user behavior trust evaluation, and then determines the reasonability of authorization. The classification of behavior characteristics involved in ACRA, in contrast to operation behavior in traditional access control, is a complex behavior [3] that integrates the utilization of cloud computing resources, network of cloud users, and operation behavior of cloud users. The utilization of cloud computing resources includes the average utilization of hard disks, memories, CPUs, bandwidth and occupied threads. The network of cloud users includes the average exception rate of login, the average propagation delay of IPs, the average time jitter of IPs and the average response time of IPs. The operation behavior of cloud users consists of the average attempt number of unauthorized operations, the average number of attacking other cloud users, the average number of illegal connections, the average number of illegally scanning important ports, the average number of running unsafe programs and the average number of escaping punishments.

3.1.2 Acquisition Methods of Behavior Trust

Analytic Hierarchy Process (AHP) deals with complex problems by breaking them into composing factors, building the hierarchy model according to these factors control relations, determining their relative importance through comparison among them, then setting the order of their relative importance under a people judgement premise. Building the judgement matrix is a key element of AHP to quantify the decision-maker thought for complex systems. However analysis found that is difficult to achieve the consistent standard. In addition, there is a difference between the consistency of judgement matrix and the consistency of human decision-thinking.

To solve the above-mentioned problem, we propose the Fuzzy Analytical Hierarchy Process (FAHP) [17, 19] that employs a Fuzzy Consistent Matrix to improve AHP algorithm.

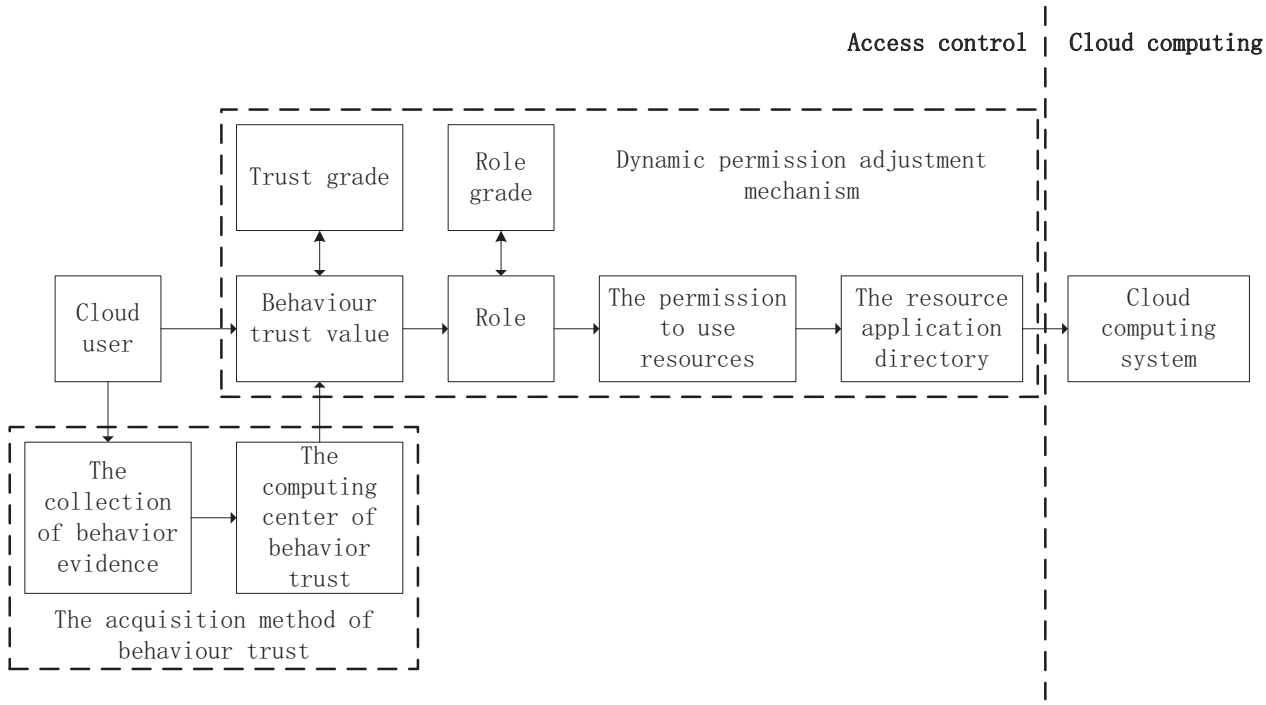


Figure 1: The overall framework of ACRA

In FAHP, the cloud users' behavior is divided into n characteristics, and each characteristic is further divided into a number of evidence types, which all have been normalized to dimensionless and increasing values on the interval [0, 1] represented with the matrix $E = (e_{ij})_{n\times m}$, where m denotes the maximum term in these characteristics, and the item that doesn't reach m should be substituted with zero.

In order to obtain the initial judgment matrix $EQ = (eq_{ij})_{m\times m}$, there m evidence $E = (e_1, e_2, ..., e_m)$, dualistic comparison of the importance of e_i and e_j in the evidence set is conducted:

$$eq_{ij} = \left\{ \begin{array}{l} 0 \text{ if } \mathrm{e_i} < \mathrm{e_j} \\ 0.5 \text{ if } \mathrm{e_i} = \mathrm{e_j} \\ 1 \text{ otherwise} \end{array} \right\} \quad (1)$$

The initial judgment matrix is transformed into a fuzzy consistent matrix $Q = (q_{ij})_{m\times m}$ [8].

Theorem 1. *If fuzzy reciprocal matrix $Q = (q_{ij})_{m\times m}$ is calculated with the following formulas, then the result is a fuzzy consistent matrix.*

$$eq_{ij} = \left\{ \begin{array}{l} \mathrm{q_i} = \sum_{\mathrm{k=1}}^{\mathrm{m}} \mathrm{eq_{ik}} \ \mathrm{i} = 1, 2, ..., \mathrm{m} \\ \mathrm{q_{ij}} = (\mathrm{q_i} - \mathrm{q_j})/2\mathrm{m} + 0.5 \end{array} \right\} \quad (2)$$

Proof.

$$\begin{aligned} q_{ij} + q_{ji} &= \frac{q_i - q_j}{2m} + 0.5 + \frac{q_j - q_i}{2m} + 0.5 \\ &= 1. \end{aligned}$$

So $Q = (q_{ij})_{m\times m}$ is a fuzzy reciprocal matrix:

$$\begin{aligned} q_{ij} &= \frac{q_i - q_j}{2m} + 0.5 \\ &= \frac{(q_i - q_k) - (q_j - q_k)}{2m} + 0.5 \\ &= \frac{q_i - q_k}{2m} + 0.5 - (\frac{q_j - q_k}{2m} + 0.5) + 0.5 \\ &= q_{ik} + q_{jk} + 0.5. \end{aligned}$$

□

Therefore $Q = (q_{ij})_{m\times m}$ is a fuzzy consistent matrix. Because the fuzzy consistent matrix has its special features that make it fit the consistency of human decision-thinking the fuzzy consistent matrix can be applied in AHP.

The weight vector $w = (w_1, w_2, ..., w_m)^T$ of a certain characteristic's evidence is calculated, where:

$$w_i = \frac{1}{m(m-1)/2}[\sum_{k=1}^{m} q_{ik} - 0.5]. \quad (3)$$

Then the assessed value matrix of cloud users' behavioral characteristics is calculated, according to the evidence matrix $E = (e_{ij})_{n\times m}$ and weight matrix $W = (w_{ij})_{n\times m}$, the value on the diagonal of the matrix obtained with $E \times W_T$ is the characteristic assessed value matrix $F = (f_1, f_2, ..., f_m)$.

The initial behavior trust value of a cloud user is de-

fined as follows:

$$T_{initial} = F \times W_f^T = \sum_{i=1}^{n} f_i w_i, \quad (4)$$

where $W_f = (w_{f1}, w_{f2}, ..., w_{fn})$ is the weight set of behavior characteristics of cloud users.

It is possible to predict the trend of cloud users' behavior trust according to the sliding window principles and historical behavior trust records. In order to achieve the average effect of historical behavior trust on $T_{initial}$, the time decay value of cloud users' behavior trust with a recording time span z is calculated as follow:

$$T_{average} = \sum_{i=1}^{z} \frac{t_{old}^{(i)}}{t_{new} - t_{old}^{(i)} + 1}, \quad (5)$$

where $\frac{1}{t_{new} - t_{old}^{(i)} + 1}$ is the time decay factor of cloud users' behavior trust $t_{old}^{(i)}$ with the recording number i,that is, the effect of historical behavior trust on current cloud users' behavior trust has become weaker with the passage of time.

Finally, according to the initial behavior trust $T_{initial}$ and the average effect of historical behavior trust $T_{average}$, the final value of cloud users' comprehensive behavior trust can be obtained with the following formula:

$$T_{final} = \alpha T_{initial} + (1 - \alpha) T_{average}. \quad (6)$$

3.2 Rules in Authorization

The purpose of obtaining the value of cloud users' behavior trust is to ensure authorization accuracy. Authorization rules involved in ACRA are different from those in traditional access control technologies. The cloud user permission refers to the right that users have to use a certain quantity of cloud resources, and the scope of permission is closely related to the behavior trust of the cloud user.

- The grading of behavior trust: In order to give a more meticulous authorization to cloud users u_i, their behavior trusts are graded into $G = (0, 1, ..., k, ..., , q)$, if $t_k \leq T(u_i) \leq t_{k+1}$ in which t_k, t_{k+1} $0 \leq k \leq q$ respectively represents the minimum and maximum value in a grade interval of behavior trust, and the trust grade of this cloud user is k.

- Mapping relationship: There is a one-one mapping relationship among the trust grade $G = (0, 1, ..., k, ..., , q)$ involved in the authorization, role $R = (R_0, R_1, ..., R_k, ..., , R_q)$ and permission $P = (P_0, P_1, ..., P_k, ..., , P_q)$. That is, after determining the trust grade of the cloud user u_i, cloud computing system will assign a grade role to the corresponding trust grade so as to authorize the user.

- Permission setting: The permission in cloud computing environment refers to the permission P of the resources utilization. It is the upper limit for the amounts of cloud resources assigned to cloud users who have applied for resource access. There is a inheritance relationship $P_0 \subseteq P_1 \subseteq ... \subseteq P_k \subseteq ...P_q$ in the permission P of the resources utilization.

  After obtaining a role, the cloud user will be granted permission to use cloud resources. The scope of cloud user's permission depends on the grade of his role. Those who get the high grade roles can utilize more cloud resources, and vice versa. Any cloud user with any role may apply for accessing a certain amount of cloud resources after obtaining permission, and may not necessarily utilize the resources after being granted the access.

3.3 The Dynamic Permission Adjustment Mechanism

Figure 2 is the flow chart of the dynamic adjustment on cloud users permission [12].

In the calculation centre of behavior trust, the FAHP algorithm is adopted to calculate and update the value of behavior trust based on the collected evidence values in real time. The cloud computing system assigns a grade role to the cloud user by judging the trust grade of the behavior trust, so as to achieve the purpose of adjusting the scope of cloud user's permission. The behavior trust value calculated when the cloud user utilizes the resources this time will directly influence his permission next time. This dynamic permission adjustment mechanism can meet the requirements of dynamic permission management in a cloud computing environment.

4 Experimental Verification

We have conducted simulation in order to verify effectiveness of the ACRA scheme.

4.1 Experimental Setup

The experiments were conducted by the use cloud computing software CloudSim, programmed with JAVA language in Eclipse development environment, MySQL database and php Study database management software.

The trust grade vector $G = (0, 1, 2, 3, 4)$ of cloud users was set in the experiment, considering the fact that behaviors of most cloud users are basically dependable or dependable in the actual resource allocation process, and just a minority of them are undependable, the interval between inter zones is respectively set as $(0.05, 0.25, 0.60, 0.80)$ following reference, and the interval between trust values is mapping with the corresponding trust grade. The relationship between the permission and trust grade is shown in Table 1.

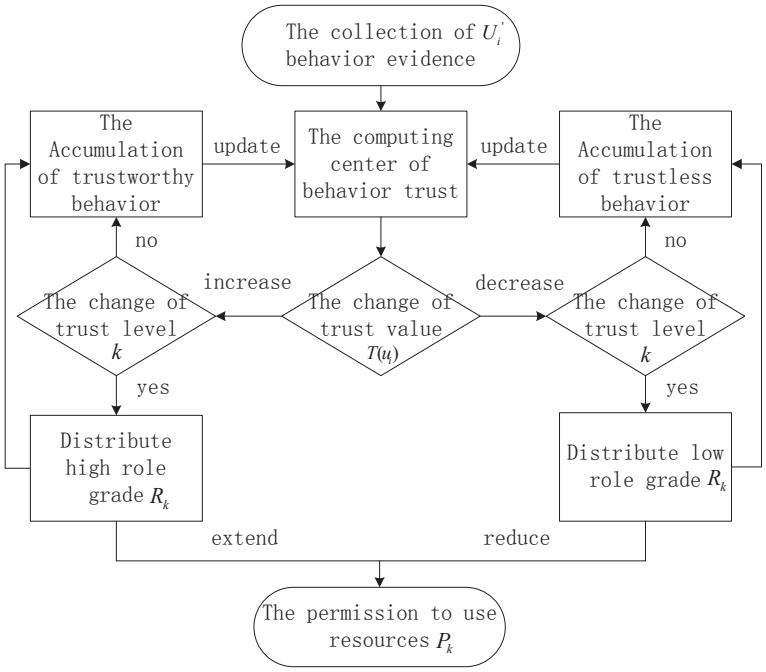


Figure 2: The flow chart of the dynamic adjustment on cloud users' permission

The record in the sliding window is $z = 5$, the historical behavior trust records before updating are shown in Table 2.

4.2 Calculation of Behavior Trust

There are three behavioral characteristics of cloud users: the resource utilization of cloud computing R, the network of cloud users N and the operation behavior of cloud users B, after basic evidence values of these three characteristics have been normalized, the average basic evidence value can be obtained as follows:

$$\begin{aligned} R &= (0.74, 0.75, 0.86, 0.63, 0.52, 0) \\ N &= (0.56, 0.49, 0.24, 0.42, 0, 0) \\ B &= (0.64, 0.77, 0.89, 0.86, 0.49, 0.91). \end{aligned}$$

The importance of R, N, B and their corresponding basic evidence in FAHP is shown as follows:

$$\begin{aligned} &R > B > N \\ &R_3 > R_2 > R_1 > R_4 > R_5 \\ &N_1 > N_4 > N_3 > N_2 \\ &B_1 > B_6 > B_2 > B_5 > B_4 > B_3. \end{aligned}$$

The weight values of all basic evidence are:

$$\begin{aligned} w &= (0.5, 0.1667, 0.3333)_T \\ w_r &= (0.2, 0.25, 0.3, 0.15, 0.1)_T \\ w_n &= (0.375, 0.125, 0.2083, 0.2917)_T \\ w_b &= (0.25, 0.1833, 0.0833, 0.1167, 0.15, 0.2167)_T. \end{aligned}$$

The initial behavior trust value $T_{initial}$ is 0.6927 according to the average basic evidence, the percentage of $T_{initial}$ in T_{final} is 0.95. Finally, the final value of comprehensive behavior trust T_{final} is 0.6588. Records of behavior trust values in the sliding window after being updated are shown in Table 3.

4.3 Simulation Analysis

We studied the behavior of a cloud user by setting a series of evidence values. Figure 3 reflects the changing trend of permission with changes in behavior trust values after the user access the cloud computing system for several times. It can be seen from the figure that when his behavior under the condition of poor performance, the number of resources who has the right to use will be decreased, when his behavior under the condition of good performance, the number of resources who has the right to use will be increased. With the change of behavior trust values, user's permission of accessing resources will be adjusted dynamically, as a cloud user utilizes cloud resources to complete computing tasks, his behaviors is closely related to the behavior trust value, which has effectively limited the user's ability in utilizing cloud resources utilization.

5 Scheme Analysis

5.1 The Safe Reliability

The scheme activates the role according to trust grade, which can ensure that a user who has successfully accessed the cloud computing system is trusted. Malicious partic-

Table 1: The classification of trust grade intervals

Trust grade	Meaning	Inter zone	Permission
0	Undependable	[0.00,0.05]	P_0
1	Low dependable	(0.05,0.25]	P_1
2	Basically dependable	(0.25,0.60]	P_2
3	Dependable	(0.60,0.80]	P_3
4	High dependable	(0.80,1.00]	P_4

Table 2: Records of behavior trusts in the sliding window before updating

ID	6	7	8	9	10
Trust	0.1367	0.3486	0.724	0.2496	0.7231
Time1	2015.01.23	2015.02.03	2015.02.07	2015.03.02	2015.04.08
Time2	18:42:11	02:33:15	05:27:50	14:12:27	15:06:14

Table 3: Records of behavior trust values in the sliding window after being updated

ID	7	8	9	10	11
Trust	0.3486	0.724	0.2496	0.7231	0.6588
Time1	2015.02.03	2015.02.07	2015.03.02	2015.04.08	2015.07.23
Time2	02:33:15	05:27:50	14:12:27	15:06:14	16:25:00

ipants will be locked out of the cloud computing system, therefore intrusion from illegal users can be prevented. By using role grade-based dynamic authorization instead of direct authorization at user registration, the proposed scheme can solve the deficiency of the traditional RBAC model, in which the administrator assigned user role may allow insiders to obtain opportunities to tamper user data. Therefore, the scheme greatly improves the cloud computing system's stability, reliability and security.

5.2 Dynamism

The traditional RBAC model is a static authorization model in which user permission is statically assigned by system administrators. This model includes two static assignment parts: user role assignment and user' role-based authorization. The scheme presented in this paper enables dynamic access control by introducing the concept of a behavior trust value into these assignment parts. The proposed scheme calculates a user behavior trust value by collecting the dynamic data, and dynamically activates the user's role, and authorizes the user's access, which gives a user different level of access permission at different time.

6 Conclusions

In this paper we studied resource allocation in the authorization stage of cloud computing and proposed a new access control-based scheme for cloud resource allocation - ACRA. This scheme first conducts a comprehensive analysis on behavioral characteristics of cloud users, and acquires user behavior trust values using Fuzzy Analytic Hierarchy Process (FAHP). The scheme then decides authorization rules for cloud users and provides a mechanism for dynamic permission adjustment. Experimental results show that ACRA can achieve more flexible and meticulous authorization, effectively restrict the ability of cloud users in resources access, provides the basis for allocating resources to cloud users with different behavior performances, improve overall cloud resource utilization and protect security of the cloud computing environment. The next goal of our research work is to consider performance overhead of the scheme model.

Acknowledgments

This work is supported by College Science and Technology Research Project of Hebei Province (ZD20131016), Natural Science Fund Project of Hebei Province (F2013208137).

References

[1] L. Chang, F. Wang, L. Zhao, Y. Jia, and Z. Cheng, "CT-RBAC: An access control model in cloud com-

puting," *Microelectronics and Computer*, vol. 31, no. 6, pp. 152–157, 2014.

[2] T. Che, J. Ma, N. Li, and C. Wang, "A security quantitative analysis method for access control based on security entropy," *International Journal of Network Security*, vol. 17, no. 5, pp. 517–521, 2015.

[3] K. V. Devi and S. Vetha, "Capacity based resource allocation in cloud," in *Proceedings of 2014 International Conference onCommunication and Network Technologies (ICCNT'14)*, pp. 24–26, Sivakasi, India, Dec. 2014.

[4] Z. Feng, Z. Qin, D. Yuan, and Y. Qing, "Key techniques of access control for cloud computing," *Chinese Journal of Electronics*, vol. 43, no. 2, pp. 312–319, 2015.

[5] W. F. Hsien, C. Yang, and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[6] Q. Li, Q. Hao, L. Xiao, and Z. Li, "Adaptive management and multi-objective optimization for virtual machine placement in cloud computing," *Chinese Journal of Computers*, vol. 34, no. 12, pp. 2253–2266, 2011.

[7] C. Ling, W. F. Hsien, and M. S. Hwang, "A double circular chain intrusion detection for cloud computing based on adjointvm approach," *International Journal of Network Security*, vol. 18, no. 2, pp. 397–400, 2016.

[8] Y. Liu and J. Zhang, "Consistency and scale in fahp," *Journal of Northeast Normal University(Natural Science Edition)*, vol. 42, no. 2, pp. 27–30, 2010.

[9] E. O. Osei and J. B. H. Acquah, "Cloud computing login authentication redesign," *International Journal of Electronics and Information Engineering*, vol. 1, no. 1, pp. 1–8, 2014.

[10] S. Parikh, "A survey on cloud computing resource allocation techniques," in *Proceedings of 2013 Nirma University International Conference on Engineering (NUiCONE'13)*, pp. 397–404, Taipei, Taiwan, Nov. 2013.

[11] M. Risch, G. Li, and C. Courcoubetis, "The gridecon platform: A business scenario tested for commercial cloud services," in *Proceedings of the 9th Int Conf on Cluster Computing and the Grid*, pp. 46–59, Berlin, Germany, Aug. 2009.

[12] N. Sklavos and O. Koufopavlou, "Access control in networks hierarchy: Implementation of key management protocol," *International Journal of Network Security*, vol. 1, no. 2, pp. 103–109, 2005.

[13] Y. Tan and C. Wang, "Trust evaluation based on user behavior in cloud computing," *Microelectronics and Computer*, vol. 32, no. 11, pp. 148–151, 2015.

[14] G. Tian and D. An, "Failure rules based node resource provision policy for cloud computing," in *Proceedings of 2010 IEEE International Symposiums on Intelligent Signal Processing*, pp. 397–404, Taipei, Taiwan, Sept. 2010.

[15] P. Varalakshmi, T. H. Judgi, and M. Fareen, "Local trust based resource allocation in cloud," in *Proceedings of 2013 Fifth International Conference on Advanced Computing (ICoAC'13)*, pp. 591–596, Chennai, India, Dec. 2013.

[16] Y. Wang, J. Yang, C. Xu, X. Ling, and Y. Yang, "Survey on access control technologies for cloud computing," *Journal of Software*, vol. 26, no. 5, pp. 1129–1150, 2015.

[17] C. Xiao and M. Chen, "Research on user behavior trust model based on ifahp in cloud computing environment," *Netinfo Security*, vol. 12, no. 0, pp. 14–20, 2015.

[18] H. Xiong, X. Chen, X. Fei, and H. Gui, "Attribute and rbac-based hybrid access control model," *Application Research of Computers*, vol. 33, no. 7, pp. 1–10, 2015.

[19] T. Yang, Y. Yuan, and M. Zhang, "Research of site selection of housing industrialization base based on fuzzy analytic hierarchy process," *Journal of Engineering Management*, vol. 29, no. 2, pp. 43–48, 2015.

[20] B. Yin, Y. Zhang, B. Fang, and W. Feng, "Cloud resource allocation method based on elastic resource adjustment," *Telecommunications Science*, vol. 11, no. 4, pp. 22–27, 2014.

[21] M. Zareapoor, P. Shamsolmoali, and M. A. Alam, "Establishing safe cloud: Ensuring data security and performance evaluation," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 88–99, 2014.

[22] W. Zhang, H. Zhang, D. Zhang, and T. Cheng, "Memory cooperation optimization strategies of multiple virtual machines in cloud computing environment," *Chinese Journal of Computers*, vol. 34, no. 12, pp. 2266–2277, 2011.

[23] J. Zhou, W. Zha, Y. Chen, and H. Zhang, "User-aware resource provision policy for cloud computing," *Journal of Computer Research and Development*, vol. 51, no. 5, pp. 1108–1119, 2014.

Junshe Wang received her B.S. degree from the Department of automation at Hebei Institute of Mechano-Electric Engineering in 1982. Now, she is a professor in School of Information Science and Engineering at HEBUST, located in Shijiazhuang, China. Her current research interests include management of network and management of computer information, etc.

Jinliang Liu is an assistant Engineer of Communication system and networks department at The 54th Research Institute of China Electronics Technology Group Corporation, he received his B.S. degree and M.S. degree from the Department of School of Information Science and Engineering at HEBUST, located in Shijiazhuang, China. His current research areas include management of network, access control technology and its applications

in cloud computing environment, etc.

Hongbin Zhang is an associate professor of School of Information Science and Engineering at HEBUST, he received his B.S. degree from the Department of automation at HEBUST in 1998, received his MS, and Ph.D. degrees from the School of Computer Science and Technology at Xidian University in 2005, 2009. His current research interests include management of network, insider threat analysis, etc.

Attack Intention Recognition: A Review

Abdulghani Ali Ahmed, Noorul Ahlami Kamarul Zaman
(Corresponding author: Abdulghani Ali Ahmed)
Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang
26300 Gambang, Pahang, Malaysia
(Email: abdulghani@ump.edu.my)
(Received Nov. 3, 2015; revised and accepted Apr. 6 & May 7, 2016)

Abstract

Sensitive information faces critical risks when it is transmitted through computer networks. Existing protection systems are still limited in their capacities to ensure network information has sufficient confidentiality, integrity, and availability. The rapid development in network technologies has only helped increase network attacks and hide their malicious intent. This paper analyzes attack types and classifies them according to their intent. A causal network approach is used to recognize attackers' plans and predict their intentions. Attack intention is the ultimate attack goal which the attacker attempts to achieve by executing various methods or techniques, and recognizing it will help security administrators select an appropriate protection system.

Keywords: Attack intention recognition, causal network approach, cyber security, network forensics

1 Introduction

Information security over a network has become more challenging due to the expansion of technologies for hacking and anti-forensics. Sensitive information should be treated confidentially in any system as it represents a high risk to the owners if exposed to the public. Information is at risk due to several factors, including human and technical errors, accidents and disasters, fraud, commercial espionage, and malicious damage [1, 2, 4].

Activities such as unauthorized access, damage to computer data or programs, obstruction of the functions of computer systems or networks, interception of data, and computer espionage are categorized as cybercrimes [7, 8, 10, 11, 17, 21]. Cybercrimes are broad in scope and are defined as attacks that involve the use of computers or networks to commit the crimes. According to [3, 4, 9], cyber-attacks can be categorized into unauthorized access, malicious code (malware), and interruption of services. Figure 1 shows common types of network threats.

Network forensics, as a part of network security, works with laws and guiding principles established in the judicial system to deal with cyber criminals. Network forensics has two approaches: reactive and proactive. Reactive network forensics is a traditional approach that deals with cybercrime cases a period of time after an attack. The reactive forensic approach consumes considerable time during the investigation phase. Proactive network forensics is a new, different approach that focuses on investigating concurrently with an attack [5, 14].

Figure 2 shows a framework of the generic process model in network forensics that splits the phases into two groups. The first group relies on actual time and includes five phases: preparation, detection, incident response, collection, and preservation. The second group relies on the post-investigation phases.

Authors in [16] also classify the first group as proactive and the second group as reactive. The proactive phases have advantages in saving time and money during investigation, as they work concurrently with the occurrence of the cybercrime. By contrast, reactive phases begin with the examination phase to integrate the trace data and identify the attack indicators. The indicators are then prepared for the analysis phase, which reconstructs the attack indicators either by soft computing or statistical or data mining techniques to classify and correlate the attack patterns. Attack intention is the ultimate goal the attacker is attempting to achieve by executing various methods or techniques of attack. Even for an expert, it is difficult to predict methods of attack. An attacker will work toward his goal through a sequence of tactical steps using sophisticated techniques to hide and cover his patterns. Attack Intention Recognition (AIR) is the process of using known attack scenarios to observe an attacker's behavior and infer his intention [19]. With the rapid developments in networking technology, attacks have become more dangerous than ever, deploying sophisticated mechanisms to hide malicious behavior. Understanding attackers' behavior will help security administrators recognize their intentions and better predict their activities.

In the following section, work related to this research is critically analyzed. This study discusses using proactive AIR methods to identify attack plans to predict future ac-

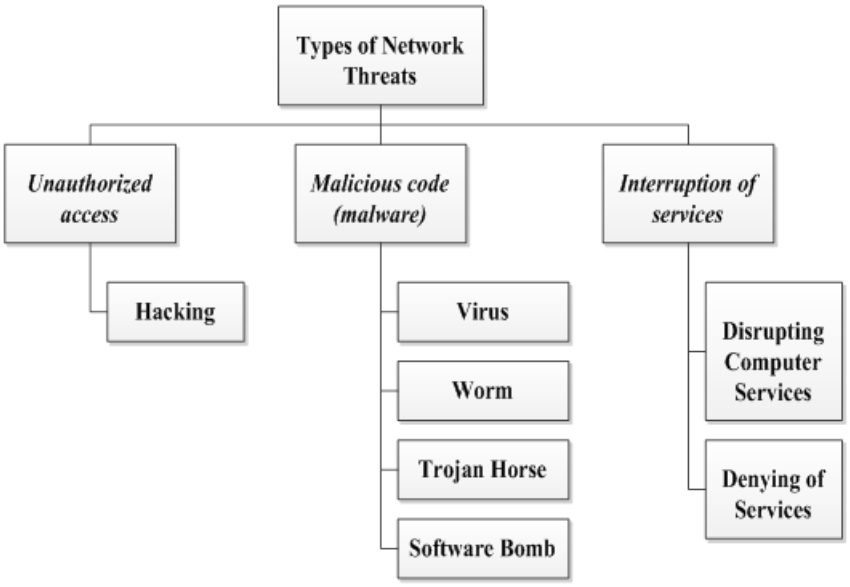


Figure 1: Common types of network threats

tions. The remainder of the paper is organized as follows. Section 2 reviews related works. Section 3 critically discusses the most relevant works, and Section 4 concludes this paper.

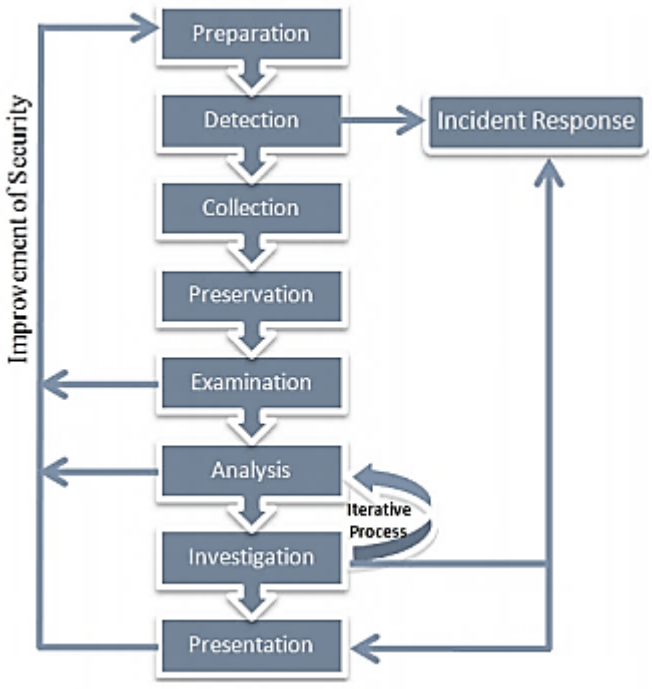


Figure 2: Generic process model

2 Related Works

Numerous studies have studied different approaches to AIR and its various methods of implementation [13, 14, 15, 18, 19, 20]. The approaches that focus on identifying attack intention are causal networks, path analysis, graphical attack, and Dynamic Bayesian Network (DBN). These approaches are described with further detail in the following subsection.

2.1 Causal Networks

The researchers in [12] studied security alert correlation, which focuses on conducting probabilistic inference to correlate and analyze attack scenarios. From the analysis, they attempted to solve other issues: (1) to identify attacker's tactics and intention and (2) to predict potential attacks. Recognizing attack plans is the process of deducing the aims of an attack from observations of its activities. Alert correlation analysis is significant for avoiding potential attacks and minimizing damage. To explicate all paths through a system which an intruder may use to accomplish his goal, attack plans or libraries are used, usually denoted by graphs. The security or vulnerability of a system is then computed by an attack tree analysis, which is based on the attacker's aims. This type of analysis can be used as a baseline for threat detection, defense, and response. However, it is a manual and time consuming process and is less scalable for a large network.

An example of an attack tree of methods for stealing and externally exporting data stored on a server is shown in Algorithm 1. The sample indicates that to obtain confidential data, an attacker may use several methods such as downloading data directly from the server or eavesdropping on the network. To gain access to a server, it is

necessary to acquire normal users' or system administrators' privileges (root).

To correlate isolated alerts, attack trees are adopted to define attack plan libraries. They are then converted to causal networks so that probability distribution can be assigned. The benefit to defining attack tree nodes by attack classes rather than specific attack is the reduced complexity of the computation for the probabilistic inference on the causal network. In implementation, a directed acyclic graph illustrates a causal network (Bayesian network), where each node symbolizes a variable with a certain set of states and directed edges denoting the cause of the dependent relationship among the variables. Probabilistic inference is applied to the causal network to evaluate goals by reviewing attack activities, thereby predicting potential future attacks.

Algorithm 1 Steal and export confidential data

```
1:  Get confidential data
2:  Get data from Server directly (OR)
3:  Get access to server
4:  Steal ID file and password file (OR)
5:  Use Trojan program (OR)
6:  Eavesdrop on the network
7:  Get System Administrator's (root) privilege
8:  Exploit Server's vulnerabilities
9:  Identify Server's OS and active ports (OR)
10: Inspect Server's activeness
11: Identify Firewall access control policy
12: Identify Firewall IP address
13: Eavesdrop on the network (OR)
14: Brute force guess
15: Eavesdrop on the network

16: Export_confidential_data
17: Transfer data via normal method (OR)
18: Transfer data via covert channel
19: Setup covert channel
```

For the test, any scenarios that have similar end goals are grouped under one evidence set due to correlated aims. This method applies attack trees to the library of attack plans. From the results observed, attack scenarios automatically correlate isolated attacks and ensure network security is controlled.

Based on [13, 15], attack intention analysis is a predictive factor for facilitating the accurate investigation of a case. This paper proposed a technique combining Dempster-Shafer (D-S) evidence theory with a probabilistic method through a causal network to predict attack intentions. The purpose of this research is to support decision making by selecting and predicting actual attack intentions and determining the best response, regardless of feasibility.

The experiment results show that the accuracy of prediction is related to the amount of evidence collected. The results also show that security can determine the highest priority value among intention probability values and make a decision that minimizes the use of time and money. However, this research has limitations. Identifying the attack intention is difficult if the malicious action is distinct from predefined attack classes. Distinguishing a deception from actual aims of attackers is also challenging. Another challenge is determining whether there is a single attacker or a collaborative group.

2.2 Path Analysis

The researchers in [18] proposed a technique that uses attack path analysis and can provide protective measures at minimum cost. Knowing an attacker's intention can help network guards make decisions as they can more easily predict potential attack paths and evaluate threats. When an attack scenario recognizes an intruder's intention, it is detailed by an attack path. Usually, successful attacks comprise a series of vulnerability exploits that grant the privileges of the projected host and use them to attack the final target. To determine the attack path on a network, the attack path on a victim host should be specified. Figure 3 shows possible attack scenarios. Note that multiple vulnerabilities can be exploited to achieve the same goal. Each attack path starts from the access node (local node) and ends at the higher privilege node.

A complete set of the possible attack paths on a victim host can be calculated using a path finding algorithm. The algorithm uses vulnerabilities, privileges, and host information to produce a graph of the attack path. A graph comprising all possible attack paths is computed once a model of the network configuration and the victim host are input. In this paper, it is assumed that an attacker will not cover his tracks after reaching his target. Generating an attack path graph requires the parameters of the host, privileges, intention, output of attack paths on a victim host on the network, and information on the network configuration.

For each network, intentions can be determined based on either the vulnerabilities and topology of the network or the focus of its business. Attention is then given to larger probability intentions. This study proposes assessing the threat by recognizing an attacker's intention and predicting the attack path. By applying the Bayesian rule, the threat situation of the entire network can be calculated when the intentions are known.

To reach the network guards' goal of protective measures at minimum cost, the minimum number of nodes is cut. Thus, an intrusive intention can be determined from the initial point using an attack path graph, which is a directed acyclic graph, to evaluate intention threat. In the experiment, intention probabilities can be computed based on the degree of difficulty in exploiting vulnerabilities. An intention capable of greater damage represents a larger value of consequence. To ensure security of the network, all intentions of attack should be blocked.

Conversely, given that attack paths remove the minimum number of nodes to disconnect the intrusive inten-

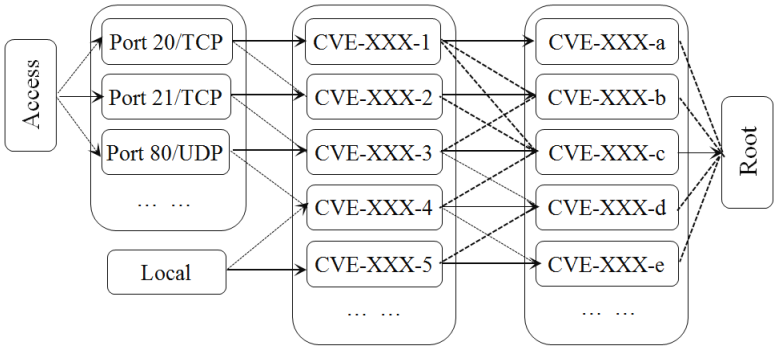


Figure 3: Framework of attack path on victim host

tion from the initial point, there is a probability that the removed nodes themselves are the target of attack. In such cases, the attack intention can go unrecognized.

2.3 Graphical Attack

The graphical model in [19] was used to recognize attack intention. The researchers attempted to verify the feasibility and validity of this method. A network security states graph, which is a directed graph, was used as a graphical model of attacks. In this model, the said graph is represented by nodes of security states that include both the states of the system and the attacker. The edges of the graph denote a relationship of state transition under the actions of attackers. No circuit is present in the graph as it is presumed that the attacker will not reintrude a host he has attacked. There is a pseudocode of algorithm that generates a network security states graph. This pseudocode shows the initial state of the network and uses available attack actions as input. To infer uncertain intentions, D-S evidence theory is used. A threat assessment is presented to evaluate the security level of a network based on the situation and the value of the intended target is determined. Figure 4 illustrates an example of a security states graph. Every "S" node is a state of network. The "H" links are hosts, and "a"s are exploitations of vulnerabilities.

Similar to the previous technique, this method also assumes that attackers have several attack plans to achieve the same intention. With D-S evidence theory, possibly every attack plan can be derived. It is useful for providing evidence and guiding decision making. The authors in [6] define attack graphs as an instrument that works out the hierarchical steps of an attack scenario by using vulnerabilities and configuration. Thus, the type attack, whether normal or anti-forensics, can be identified. Anti-forensics, as described in this paper, uses methods such as deleting system logs after hacking into a computer to prevent tracking by authorities. Using the attack evidence graph, the existence of anti-forensics attacks can be determined. The tools and techniques used by the attacker can also be identified. However, with the current mechanisms used in anti-forensics, system configuration and vulnerability information are not enough to trace the path. This is because security depends on vulnerability data but attackers use anti-forensics to hinder this action. Moreover, this approach only aims to identify the intention of the unauthorized access to a network or host that an attacker may compromise. Thus, attackers with privileged access to network are an identified challenge in this approach.

2.4 Dynamic Bayesian Network

As discussed in [20], the Dynamic Bayesian Networks (DBN) method is proposed for identifying intrusion intention. This research aims to improve on the limitation of current Intrusion Detection System (IDS) technology, which fails to apply a logical relationship between attack events. DBN is a technique for combining a static Bayesian network and a timestamp to form a new probabilistic model from the removal of order data. Figure 5 shows the DBN based on the intrusion intention identification model: (a) prior network, (b) transfer network, and (c) DBN model in time.

For the scenarios, given that a large aggregation of training data are available, the Markova Assumption is used to assume the attack goal, depending only on intentions observed under restrictions plus the last completed goal and the latest attack behavior. The process in reaching the final attack goal, based on intrusion alarm messages, is shown in Figure 6.

The experiment assumes the goal with the most probability is the final attack goal of the intruder. In this process, the final target is identified when the attacker compromises another target first to gain privilege. The disadvantage of this approach is its dependency on the last completed goal and latest attack behavior.

3 Related Work Analysis and Discussion

This section compares the related works and analyzes their models. From the discussion above, it may be

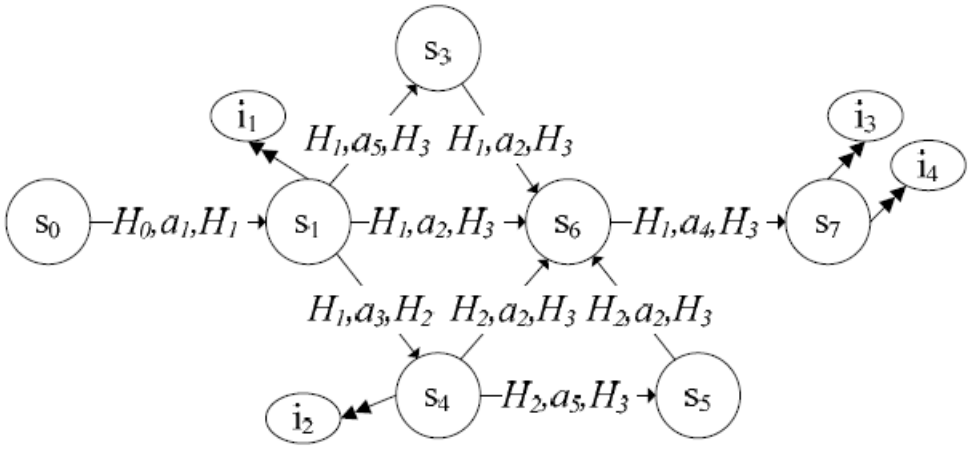


Figure 4: Example of security states graph

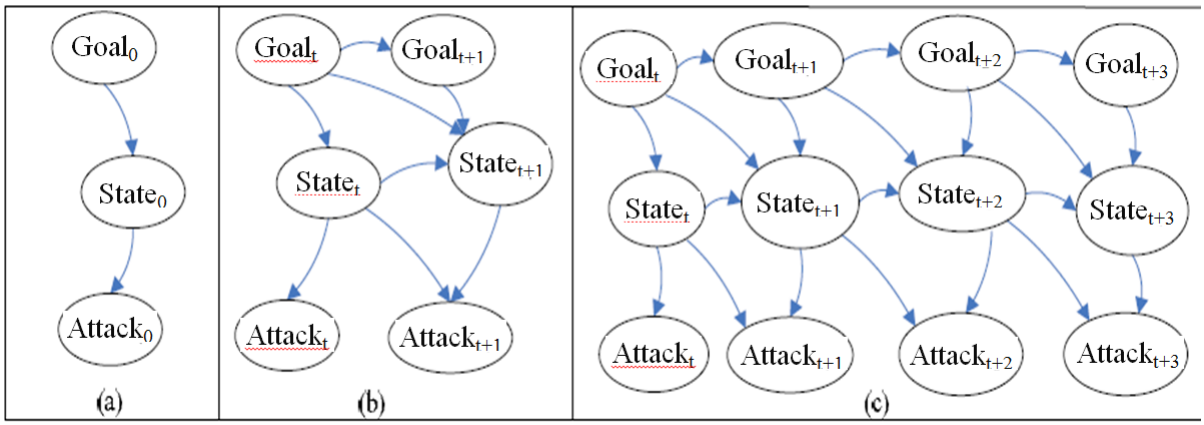


Figure 5: Dynamic Bayesian network architecture

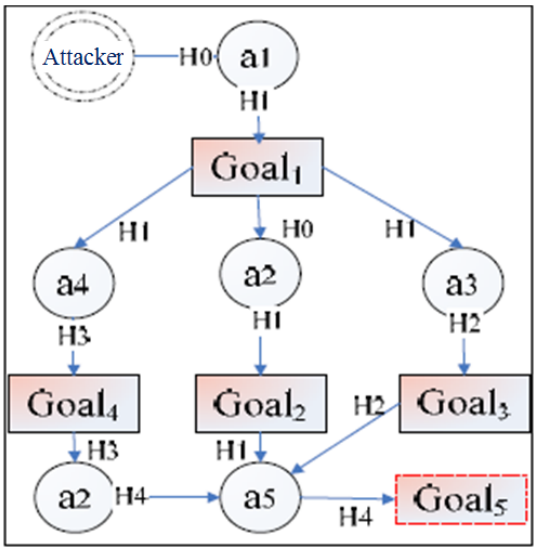


Figure 6: Process of reaching the final attack goal

observed that there are similar methods used in different models such as D-S evidence theory, Bayesian rule, and directed acyclic graphs. D-S evidence theory focuses on uncertainty to conclude the intention of an attack [22]. Bayesian rule applies probabilistic reasoning for threat assessment or determining the goal of the intrusion. Directed acyclic graphs track attacks. Directed acyclic graphs track attacks using several methods such as attack path, attack tree, or attack plan. However, attack trees have some drawbacks. They are manual processes, time consuming, and are limited to the attack plans in the library [12]. That said, the library can be expanded through the participation of security experts. Besides competence in attack recognition, the other advantages of the aforementioned approaches are discussed. Graphical models use network security states graphs. The algorithm proposed infers intent and conducts threat assessment. Similar to the graphical approach, causal networks also use graph-based techniques to correlate isolated attack scenarios after observing their relationships in attack plans. It is proposed for pinpointing attack plans and predicting upcoming attacks. However, causal networks

have an added value: by applying probabilistic inference to evaluate the likelihood of attack goals and forecast upcoming attacks based on causal networks converted from attack trees. An attack path analysis model approach to constructing attack path graphs can also recognize the intrusive intention and simultaneously calculate the threat of intention. This approach can find protective measures at minimum cost with the theory of minimum cut. Moreover, a DBN adopts probabilistic reasoning for estimating an attack. This technique can identify the intrusion intention with various alarm messages and predict incoming attacks in real-time. That said, each of the aforementioned approaches has certain limitations. These limitations are summarized in Table 1.

Table 1: Disadvantages of attack intention recognition models

Model	Limitations
Causal network	• If malicious actions are different from the predefined scope of attack, it is hard to identify them. • It is difficult to distinguish deception and actual plans of attackers. • It is difficult to determine whether the actual number of attackers.
Attack path	• Only presents the first step toward identifying intrusive intention.
Graphical	• Only presents the first step toward identifying intrusive intention.
Dynamic Bayesian networks	• Given that the attack assumption is based on the latest action, it will not work in a case of uncertain attack.

Although attack path analysis, graphical model, and causal network approaches all apply graphs in their methods, causal networks have another added value in that they compare attack path analyses and graphical models. Besides providing graph-based techniques to correlate isolated attack scenarios, they apply probabilistic inference to evaluate the likelihood of attack goals and forecast upcoming attacks based on causal networks converted from attack trees. Thus, the causal network approach will be adopted to solve the problems in this research.

4 Conclusion

This paper reviews various approaches toward attack intention recognition, including causal networks, path analysis, graphical attack, and DBNs with Markova assumptions. These approaches are all interrelated, differing from each other due to the aims of researchers. Basing on the review performed on the existing works and the critical analysis of their advantages and disadvantages, we conclude that using a causal network approach is effective for detecting network attacks that have similar intentions. For future study, an experiment will be performed to evaluate the efficiency of detecting an attack's intention. This can entail testing various methods for detecting attack intentions and seeing how each method performs in a true lab environment under real world scenarios.

Acknowledgments

RDU grant number RDU1403162, Faculty of Computer System & Software Engineering, Universiti Malaysia Pahang supported this work.

References

[1] A. A. Ahmed, A. Jantan, and M. Rasmi, "Service violation monitoring model for detecting and tracing bandwidth abuse," *Journal of Network and Systems Management*, vol. 21, no. 2, pp. 218–237, 2013.

[2] A. A. Ahmed, A. Jantan, and T. C. Wan, "Sla-based complementary approach for network intrusion detection," *Computer Communications*, vol. 34, no. 14, pp. 1738–1749, 2011.

[3] A. A. Ahmed, A. Jantan, and T. C. Wan, "Real-time detection of intrusive traffic in qos network domains," *IEEE Security & Privacy*, vol. 11, no. 6, pp. 45–53, 2013.

[4] A. A. Ahmed, A. Jantan, and T. C. Wan, "Filtration model for the detection of malicious traffic in large-scale networks," *Computer Communications*, vol. 82, no. 59-70, pp. 15–23, 2015.

[5] A. A. Ahmed, A. S. Sadiq, and M. F. Zolkipli, "Traceback model for identifying sources of distributed attacks in real time," *Security and Communication Networks*, 2016.

[6] R. Chandran and W. Q. Yan, "A comprehensive survey of antiforensics for network security," *Managing Trust in Cyberspace*, pp. 419–447, 2013.

[7] T. W. Che, J. F. Ma, Na Li, and C. Wang, "A security quantitative analysis method for access control based on security entropy," *International Journal of Network Security*, vol. 17, no. 5, pp. 517–521, 2015.

[8] B. B. Gupta, R. C. Joshi, and M. Misra, "Ann based scheme to predict number of zombies in a ddos attack.," *International Journal of Network Security*, vol. 14, no. 2, pp. 61–70, 2012.

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

[10] C. C. Lee, M. S. Hwang, and I-En Liao, "On the security of self-certified public keys," *International Journal of Information Security and Privacy*, vol. 5, no. 2, pp. 55–62, 2011.

[11] 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.

[12] X. Qin and W. Lee, "Attack plan recognition and prediction using causal networks," in *20th IEEE Annual Computer Security Applications Conference*, pp. 370–379, 2004.
[13] M. Rasmi and A. Jantan, "Aia: Attack intention analysis algorithm based on D-S theory with causal technique for network forensics- a case study," *International Journal of Digital Content Technology and its Applications*, vol. 5, no. 9, pp. 230–237, 2011.
[14] M. Rasmi and A. Al-Qerem, "Pnfea: A proposal approach for proactive network forensics evidence analysis to resolve cyber crimes," *International Journal of Computer Network and Information Security*, vol. 7, no. 2, pp. 1–25, 2015.
[15] M. Rasmi and A. Jantan, "Attack intention analysis model for network forensics," in *Software Engineering and Computer Systems*, vol. 2, pp. 403–411, Springer, 2011.
[16] M. Rasmi, A. Jantan, and H. Al-Mimi, "A new approach for resolving cyber crime in network forensics based on generic process model," in *The 6th International Conference on Information Technology (ICIT'13)*, pp. 45–53, 2013.
[17] S. Saurabh and A. S. Sairam, "Increasing accuracy and reliability of ip traceback for ddos attack using completion condition," *International Journal of Network Security*, vol. 18, no. 2, pp. 224–234, 2016.
[18] P. Wu, Y. Shuping, and J. Chen, "Recognizing intrusive intention and assessing threat based on attack path analysis," in *IEEE International Conference on Multimedia Information Networking and Security (MINES'09)*, vol. 2, pp. 450–453, 2009.
[19] P. Wu, Z. Wang, and J. Chen, "Research on attack intention recognition based on graphical model," in *Fifth IEEE International Conference on Information Assurance and Security (IAS'09)*, vol. 1, pp. 360–363, 2009.
[20] Q. Wu, R. Zheng, G. Li, and J. Zhang, "Intrusion intention identification methods based on dynamic bayesian networks," *Procedia Engineering*, vol. 15, pp. 3433–3438, 2011.
[21] Z. Yunos, R. Ahmad, and N. A. M. Sabri, "A qualitative analysis for evaluating a cyber terrorism framework in malaysia," *Information Security Journal: A Global Perspective*, vol. 24, no. 1-3, pp. 15–23, 2015.
[22] Y. Zhang, L. Yu, and W. Li, "Research of ds evidence method in network attack intention recognition," in *2nd International Conference on Electronic & Mechanical Engineering and Information Technology*, pp. 2325–2328, Atlantis Press, 2012.

Abdulghani Ali Ahmed biography. Abdulghani Ali Ahmed is a senior lecturer in the Faculty of Computer Systems & Software Engineering at Universiti Malaysia Pahang. His research interests include information security, digital forensic and cybercrimes investigation, MPLS technology, QoS and embedded real-time systems. Ahmed received the PhD in network security from Universiti Sains Malaysia in 2014. He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the International Association of Engineers (IAENG). Contact him at abdulghani@ump.edu.my.

Noorul Ahlami Kamarul Zaman biography. Noorul Ahlami Kamarul Zaman received the Bachelor in computer systems & networking with honors from University Malaysia Pahang in 2014. She received the Master in computer networking from University Malaysia Pahang in 2016. Her area of interest includes cyber security and network forensics.

Accountability in Cloud Computing by Means of Chain of Trust

Dipen Contractor, and Dhiren Patel
(Corresponding author: Dipen Contractor)
Computer Engineering Department, NIT Surat, India
(contractor.dipen@gmail.com, dhiren29p@gmail.com)
(Received Nov. 17, 2015; revised and accepted Feb. 11 & Mar. 6, 2016)

Abstract

Cloud computing offers various services in form of infrastructure, platform, and software to meet the consumer requirements. It is radically changing how information technology services are created, delivered, accessed, and managed. However, this swift has prompted concerns regarding security and privacy due to cloud computing characteristics such as the multi-tenancy, elasticity, and layered architecture. One of the major challenge is to offer accountability in cloud services across all the dependencies. When one entity relies on other entities for functioning, it creates a dependency in system and makes it difficult to sort out the responsible entity among them. In this paper, we analyze the problem of creating accountable cloud services. We utilize basic functionality provided by Trusted Computing Group (TCG) in form of chain of trust (CoT) by securely recording identities (of entities). We propose a solution that modifies existing chain of trust to build accountable cloud computing. We explore dependency relationship in building reliable chain of trust in cloud and define it for better implementation.

Keywords: Accountability, chain of trust, cloud computing, dependency

1 Introduction

Cloud computing is an amalgamation of technologies like service oriented architecture (SOA) and virtualization, that turns Internet into service delivery infrastructure. Service providers can lease a set of resources from cloud infrastructures to provide their software as services in an economical way without owning physical infrastructures [3].

Various Cloud service models serve as forms of abstraction and eliminate the need to deal with internal details of the operation, management, and state of the underlying infrastructure [27]. The cloud service provider's (CSP's) computing resources are pooled to serve all consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned according to consumer demand. The customer generally has no control of the location of the allocated resources. As a result, establishing accountability in distributed and layered architecture is an issue. The problem arises when you consider that the application is dependent on functioning of entities in order to continue processing, and thus a single entity failure could stop the entire application.

According to Cloud Computing Incidents Database (CCID), major CSPs have suffered downtime ranging from a few minutes to a few hours [10, 30]. During a cloud service disruption, affected customers will not be able to access the cloud service and in some cases may suffer degraded performance. For example, in June 2013, major cyber-attack was launched on North Korea by South Korea [20, 23]. The attack compromised an update of application hosted at cloud service provider that also hosts North Korean government websites. Many government websites were defaced by this attack. According to Trend Micro [24], website defacement was only the tip of the iceberg; sensitive information (of military and government) was also compromised.

Trusted computing architecture [26] offers a concept of a chain of trust (CoT). We explore this concept using few additional operations to model dependencies in distributed and layered architecture of cloud. Accountability of chain is rooted from tamper resistant hardware and identity (and integrity) of each component running on a particular platform can be assured. Propagation of chain follows the principle of measure before loading [26]. It means that the entity that is executed; measures the identity of the next entity (to be executed), then passes on the execution to the measured entity. This functionality is associated with SElinux, which provides isolation of execution of any program with LSM (Linux security module) hooks [11]. According to David et al. [14], any mechanism that promotes accountability should have following two basic features:

1.1 Tamper-resistance

A mechanism to promote accountability should deter and detect any modification or malfunctioning in it [6]. Mechanism must be tamper resistant or at least tamper evident. No entity can bypass assessment operations, and if it tries, it can be identified easily. Consumer and provider both can rely on such mechanism and present it as a proof to any third party (if dispute arises). In fault detection, one can decide responsibility and act accordingly.

1.2 Privacy and Transparency Balance

Accountability promotes control and transparency in system [15]. Keeping record of all entities brings transparency, however; this may help the attacker to launch specific attack or leads rivals to know insights of cloud. Maintaining privacy while recording identities will equalize transparency.

Considering dependence relationships in cloud computing with features mentioned above, we present formalization of Chain of Trust applicable with service level agreement or third party certificates.

Rest of this paper is organized as follows: Section 2 discusses cloud computing basics. In Section 3, we discuss dependencies in cloud scenario. In Section 4, we discuss chain of trust and its formation in cloud computing. In Section 5, we present formal representation of chain of trust with conclusion and references at the end.

2 Background

National Institute of Standards and Technology (NIST) has defined the Cloud Computing as [12].

A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

2.1 Service Delivery Models

The Service model describes an organization's scope and control over the computing environment and characterizes the level of abstraction for its use. As shown in Figure 1, three well-known and often used service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

Cloud computing is not restricted to Infrastructure/Platform/Software as a Service; it can be further extended to provide a variety of service models. Armbrust et al. [4] coin the phrase "X as a Service (XaaS)"; where X can be anything like data, management, security etc. that can be provided to consumer as a service.

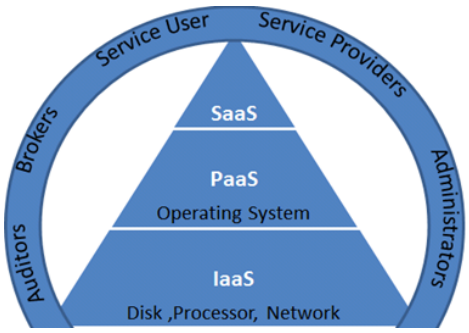


Figure 1: Cloud computing with layers

2.2 Cloud Actors

The NIST cloud computing reference architecture recognizes the main actors in a cloud ecosystem, their activities and functions in terms of cloud computing.

Service Consumer: A service consumer is a person or an organization that uses service from, one or more cloud service providers.

Cloud service provider (CSP): A cloud service provider is an organization, or entity responsible for making a cloud service available to interested parties.

Service provider (SP): A service providers is an organization, or entity responsible for building or combining individual services such as IaaS, PaaS, or SaaS.

Cloud auditor: A cloud auditor is a party that can perform independent assessment of services, system operations, performance, and security of the cloud implementation.

Cloud broker: A cloud broker is an entity that manages the use, and delivery of cloud services, and/or negotiates relationships between cloud service providers and cloud consumers.

Cloud carrier: A cloud carrier is an intermediary (actor) that provides connectivity of cloud services from providers to cloud consumers.

3 Dependencies

Cloud Computing environment consists of a number of players (actors) that interact in fragile manner, to benefits for their own and for others. Individual service providers can independently manage policies, and controls cloud entities. Cloud computing ecosystems enable highly dynamic and effective organizational collaborations. Organizations (dispersed geographically) can provide services from different levels of abstraction (e.g. business, architecture, or programming). These abstractions create dependencies.

Many researchers have found different types of dependencies in cloud computing [16, 28], such as *organizational*

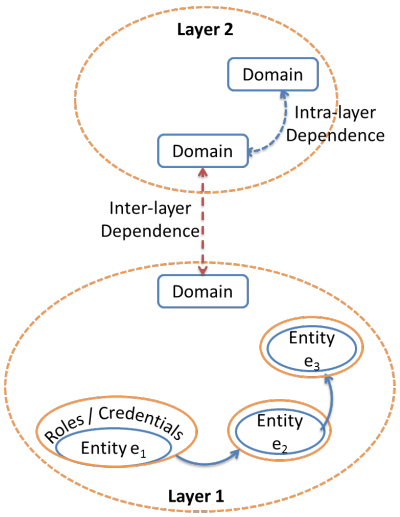


Figure 2: Conceptual model for architectural dependencies in cloud computing

and *architectural*. Organizational dependencies further classified (based on their existence) as *inter-layer* and *intra-layer* dependencies. According to Siani Pearson et al. [16] organizational dependencies may arise in situation where cloud service is composed from different services provided by different service providers. Due to that, accountability of system is shared with a cloud provider as well as with individual service providers.

Consumers must understand the scope of system management and monitoring, including access management, change management, configuration management, patch management, and vulnerability management of individual service providers.

3.1 Architectural Dependencies

We conceptualize the architecture of cloud computing to elaborate architectural dependencies. We try to intricate entity, service, domains, and layer for understanding dependence relationship.

As depicted in Figure 2, Domain represents a grouping of similar entities inside a layer. E.g. infrastructure (IaaS) security domain includes physical access control mechanism for physical resources. Each domain contains a predefined policy [21] based on that; rules, credentials or attributes are assigned to each entity. An Entity can be defined as physical resource (e.g. memory or disk), a process, or services in cloud computing. Functionality of an entity depends on other entities as shown in Figure 2. Entities from different domain communicate through layers. Entities from different domains and layers can be accounted in a single chain. The dependence relation is the relation that exits between entities of different domains across layers. Each domain operates with different policy so it is essential to handle dependence relation carefully in chain construction.

3.2 Organizational Dependencies

This section intends to support a discussion of accountability aspects of cloud computing by presenting simple usage scenarios from client's perspective. Depending on the deployment model (i.e. private, community, public, and hybrid); cloud providers, and users interact differently. Traditionally client server architecture does not have dependency relationships but in a cloud like environment, it could be between layers (SaaS, PaaS and IaaS). Evolving public cloud services are complex and dependent on providers and provider to provider as connections. In fact, the SaaS service you receive may be provided by another IaaS provider [13].

To elaborate the situation, we present highly outsourced scenario of cloud computing. We assume a cloud service provider borrows platform from PaaS provider to host applications of different software providers. The PaaS provider might have leased infrastructure from public or private IaaS provider. As shown in Figure 3, the situation is similar to hybrid cloud computing. The main issue lies in the form of establishing accountability. For example; if a cloud service consumer complains about malfunctioning of a particular service, then how cloud provider will come to know which layer or domain has a fault?. Assuming that cloud service provider has identified a particular service provider; Since ownership of the infrastructure belongs to another service provider, it is difficult for cloud service provider to investigate without proof. The solution to this problem lies in securely keeping identities of all the entities involved from different layers or providers.

4 Accountability in Cloud Computing

Accountability is about defining governance to comply in a responsible manner with internal and external criteria, ensuring implementation of appropriate actions, explaining and justifying those actions and remedying any failure to act properly [15].

Accountability and its different attributes for distributed dependable systems are briefly studied by Siani Pearson [16] and other researchers. They have worked in A4cloud project [1] for promoting accountability in cloud architectures. Accountability could be divided in two types; prospective accountability (preventive controls) and retrospective accountability (detective controls). Detective controls for the cloud include secure and trustworthy auditing, tracking, reporting, and monitoring of system.

Main components of this accountability notion are transparency, responsibility, provision for assurance, and satisfaction of obligations [14].

For the purposes of this paper, accountable cloud maintains a tamper-evident record that provides non-repudiable evidence. Based on this record, a faulty node

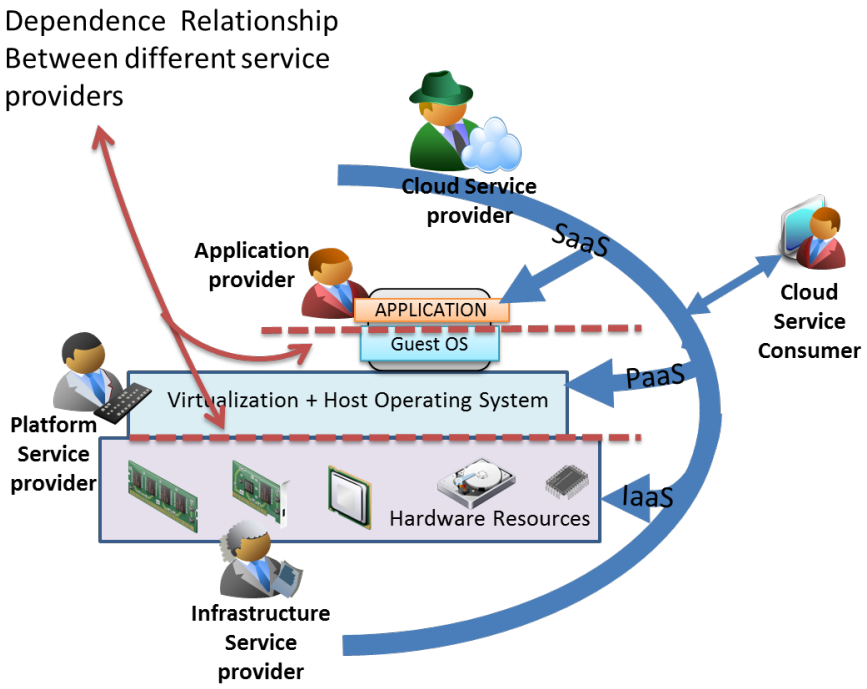


Figure 3: Cloud computing services outsourced from different providers

(whose observable behavior deviates from that of a correct node) can be detected. Accountable cloud provides primitives for cloud carriers to validate the identities of entities associated in cloud.

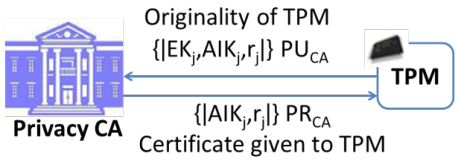


Figure 4: Certificate issued to TPM from privacy CA

We setup a chain of trust that could be fully embedded in to each layer. This implies that the consumer needs to know all information regarding identities of entities involved in service orchestration [29]. To provide tamper evidence feature, we also maintain original hash of individual modules as they load to identify mismatched entity. Individual service providers are accountable for their own layer activity. Hence, consumers should understand the dependency of their application on all services and assess risks pertaining to third-party service providers. CSPs have been reluctant to share information relating to platform security using the argument that it could provide insights to hackers. However, consumers should demand transparency from CSPs and seek information necessary to perform risk assessment and ongoing security management. Before utilizing services from any CSP, consumer can ask about all identities of all the entities and service providers involved in that services.

5 Chain of Trust

As explained earlier, we need to keep track of all the entities that are involved in service creation at each layer. A chain of trust is a term used to describe the sequence of hashes that incorporates different entities that spawns over multiple layers in a cloud [19].

The first element of the chain (Root) should be reliable and it can vouch for its accountability (e.g. IBM's 4758 secure processor [8] and a tamper-evident hardware chip [22]). During the initialization of platform, Root entity is loaded first, and then other modules are loaded. The Root records identity (in terms of hash) of the second element after booting of the platform and continue to build the chain. The second element then records identity of the third element in the chain within the layer. As the second element is already assessed, it assesses the third element's integrity, and so on. These hashes will be securely sent to consumer (or trusted third party) for verification. With availability of actual hash from original manufacturer and reference database [25], one can easily identify mismatched entity.

Chain of trust concept technically relies on TCG (Trusted Computing Group) architecture for recording hash and reporting of it using cryptographic primitives. To this effect, TCG specifies a hardware module, the Trusted Platform Module (TPM) [18]. TPM is a tamper resistant piece of cryptographic hardware built onto the system board. It implements primitive cryptographic functions, using which more complex features can be derived.

The manufacturer embeds a unique master keypair

named as endorsement key (EK) during the creation of TPM chip. The private part of EK never leaves out of the chip. It also embeds mechanism [2] for a certificate on the public key-part of EK, which vouches for the authenticity of TPM. This certificate allows a third party or consumer to verify that messages signed with this EK come from a genuine TPM. Moreover, it allows a third party to communicate over a trusted channel with the TPM. However, due to privacy concerns, the EK is not used directly but an intermediary attestation identity key (AIK) is used, which is wrapped in a certificate, signed by an externally trusted certificate authority (CA) (as shown in Figure 4).

Verification of CoT requires certificate given by any trusted third party; in our case Privacy CA. This certificate assures involvement of accountable service provider. Moreover, a certificate $Certi_j(AIK_j, r_j)$ also contains hash of root entity r_j, so that it can be verified. Storing hash of entities will require large and secure storage. TPM comes with a limited number of registers named platform configuration registers (PCRs). Also at application layer, two lists are maintained viz, stored measurement list (SML) and integrity measurement list (IML) [9]. To permit a TPM version to perform in the cloud, specifications have been generated for a virtual TPM (vTPM) [17] that provides software instances of TPMs for each virtual machine. As shown in Figure 7, a chain is built across layers with a single root (as in a private cloud deployment environment). In other case, highly outsourced cloud would have individual root for each layer, so we name it as multi-rooted chains. Working of CoT is explained with few basic operations as discussed below, which combines various functionalities of TPM.

5.1 Extend Operation

As defined in reference architecture of TPM [26], extend operation maintains the final single hash of a platform. It discards individual hashes after adding to a single value.. Verifier has to derive all the steps and try to get that single hash value. As explained in previous section, hashes of individual identities are sequentially stored. For identification and verification of dependencies in CoT, we have changed traditional extend operation as shown in Figure 5. CoT also maintains original hash of the modules of entities as evidence and provided when explicitly asked by the consumer or third party. Extend operation stores actual hash and extended hash of individual entities to different list. For operating on PCRs, only TPM can write or extend it. SML and IML are utilized for storing hashes encrypted with AIK at application level and sent whenever they are required.

CoT comprises identities of entities and actors involved in cloud service life cycle with privacy. Let us define entity set $E = \{e_1, e_2, e_3, e_4, \ldots, e_n\}$. Hash of the each entity has one to one mapping to set $H = \{h_1, h_2, h_3, \ldots, h_n\}$. Each entity belongs to a particular domain from set $D = \{d_1, d_2, d_3, \ldots, d_n\}$. Each domain operates at a particular layer of set $L = \{l_1, l_2, l_3, \ldots, l_n\}$. Mainly cloud standard architecture is based on SPI (software, platform, infrastructure) framework [31] and that contains only three layers. To define dependence relationship, we need to define dependency in cloud. We begin with platform configuration registers. Root represents a first element in chain whose hash r1 is recorded in initial register.

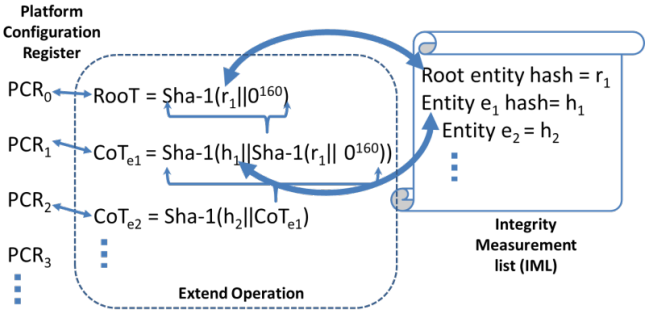


Figure 5: Extend operation in CoT: Our proposal

$$PCR0 = Root = SHA1(r_1 \parallel 0^{160}).$$

Subsequently;

$$\begin{aligned} PCR1 &= CoT_{e1} = SHA1(h_1 \parallel SHA1(r_1 \parallel 0^{160})) \\ PCR2 &= CoT_{e2} \\ &= SHA1(h_2 \parallel SHA1(h_1 \parallel SHA1(r_1 \parallel 0)) \\ &= SHA1(h_2 \parallel CoT_{e1}). \end{aligned}$$

5.2 Dependency Operation

Dependency is expressed by both extending and hashing, symbolized by dependency operator Π. The functionality of entity e_2 is dependent on e_1, and it can be represented as

$$Dep_{e2} = (e_2 \Pi e_1) = SHA1(h_2 \parallel CoT_{e1}).$$

In similar way, individual domain's CoT can be formalized as

$$CoT_{d1} = Dep_{en} = (e_n \Pi e_{n-1} \Pi e_{n-2} \Pi \ \cdots \Pi e_1).$$

As described earlier, multiple domains are contained in a layer, so layer's dependency in a private cloud scenario (single rooted chain) can be shown as

$$CoT_{l1} = Dep_{dn} = (d_n \Pi d_{n-1} \Pi d_{n-2} \Pi \cdots \Pi d_1).$$

5.2.1 Dependency Relation

As explained in architectural dependency section, dependency relation exits between two different layers (or domains) which may be owned by different service providers and therefore there exits multi-rooted CoT.

Typically, for cloud service provider, it can be represented as,

$$\begin{aligned} CoT_{csp} &= Dep_{ln} \\ &= (l_n \Pi l_{n-1} \Pi l_{n-2} \Pi \ldots \Pi l_1) \\ &= (l_{SaaS} \Pi l_{PaaS} \Pi l_{IaaS}). \end{aligned}$$

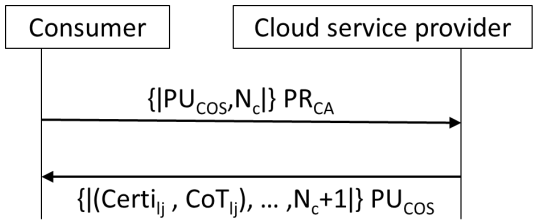


Figure 6: Transfer of CoT

Cloud service provider sends an individual layer wise CoT with its certificate which is used for authenticity and verifiability. Formally, certificate of a layer can be $Certi_{lj} = AIK_j, r_j$ where j is a number of that layer.

Here, AIK proves identity of a service provider and r (hash of the root) can be used for verification of CoT while maintaining privacy. The extend operation preserves the order of dependency; an entity cannot pretend to occur after a certain event as ordering is automatic. Numbers of PCRs are limited on TPM chip so SML will be utilized afterward. Hashing reduces the amount of data that needs to be stored, and extended in order to detect manipulation.

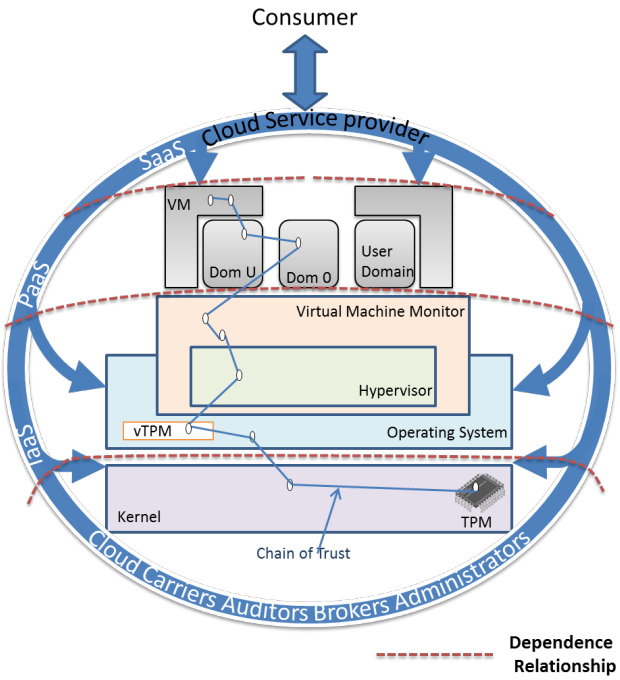


Figure 7: Cloud computing and CoT

5.2.2 Verification of CoT

Verification of CoT will be done at consumer side, but it may be delegated to third party based on computational powers. TPM works well with asymmetric key cryptography; while keeping in mind adversary present on the network. Initially, consumer sends a certificate containing public key PU_{COS} and nonce N_c (Nonce is used to ensure freshness of certain responses), given by trusted third party CA. Then CSP will reply all individual CoT of layers (denoted by j) with its certificates, encrypted with consumer's public key.

Our formal model creates a single chain of trust that can accommodate different roots and handle dependence relationships. Verification of this chain can be done by trusted third party or even at consumer side (with the help of reference manifest database) [25]. These reference hashes are collected from the original source: i.e. the software and hardware manufacturers. Each certificate provides identity of a service provider. After matching all the hashes of CoT, Root hash notifies the completeness of the chain to consumer.

5.2.3 Implementation of CoT in Cloud Computing

In our experiments; we use host machine with Ubuntu 12.04 and Xen 4.3.0 hypervisor [5] based cloud test bed with various domains. Domain-0 is the highest privileged domain; consumer operates at individual domain-U. Without enabling TC (TPM chip from BIOS), we initiated domain-0, then we compiled a user kernel and from which we created our Master domain for TC. We then enabled TC from BIOS. We kept minimal functionality and less interfaces for this domain. Now, from this domain, we can initiate individual domain CoT. Each domain-U receives a vTPM instance for integrity measurement. Consumer can ask for complete CoT (from infrastructure entity resources to SaaS resources).Therefore, this CoT approach is useful to both parties viz; CSP and consumers. CSP can rectify a fault and decide responsibility, and end-user can present it as a proof for remediation [15].Currently we have implemented it using basic scripting language i.e. python. From our previous work [7], we utilized communication mechanism to get individual chain from different domains. Actual PCR values and its corresponding CoT values are shown in Figure 8.

6 Conclusion

Everything as a service concept of cloud environment allows easier utilization of resources of different providers but it makes difficult to establish accountability of low-level entities. We propose chain of trust (CoT) as one solution to provide recording, transferring, and verifying identities of entities. Offering transparency while maintaining privacy is achieved with CoT. Verifying individual terms will lead to tamper evidence property of CoT. Secure generation of keys and certificate denote tamper resistance nature of system and thus CoT could be an acceptable solution to manage and verify architectural and organization dependencies present in cloud computing.

Extend Operation

No.	Component's Hash value	Component name
c1	298df125b260ef64201bdf0815c003873eedd50e	[BIOS:EV_S_CRTM_VERSION]
c2	B794d4c2eddf03b69bdc8ac0a1cbf9278fe131ca0	[BIOS:EV_POST_CODE(EV_CODE_NOCERT)]
c3	878e8aae5975ed177335c2cc60685dbeec04af17	[BIOS:EV_EFI_PLATFORM_FIRMWARE_BLOB]
c4	9069ca78e7450a285173431b3e52c5c252990473	[BIOS:EV_SEPARATOR, 00000000]
c5	4e0209993c236720356703f4f66b7cb4439710cb	[BIOS:EV_S_CRTM_CONTENTS]
c6	060b13c687d713e9fa2d0d0a8ef51a05ade90a2c	[BIOS:EV_S_CRTM_CONTENTS]

PCR_0	**66 3F 8E E7 5A 5C 33 FD EE 81 AF 20 FA 9E 7F 65 1F 47 91 35**

Similarly , We received Following values

PCR_1	**BB C0 45 C1 9F 21 F5 4F 49 B1 38 95 EC 49 A9 EC BF 15 F8 1D**
PCR_2	**0C E8 06 1E BA 5C 53 2C 62 59 5D DB 30 75 CA E8 A8 57 53 61**
PCR_3	**B2 A8 3B 0E BF 2F 83 74 29 9A 5B 2B DF C3 1E A9 55 AD 72 36**
PCR_4	**51 D5 3B C6 51 32 A5 76 55 20 7B 5D B3 24 42 7C A6 47 3D FA**
PCR_5	**45 A3 23 38 2B D9 33 F0 8E 7F 0E 25 6B C8 24 9E 40 95 B1 EC**
PCR_6	**B2 A8 3B 0E BF 2F 83 74 29 9A 5B 2B DF C3 1E A9 55 AD 72 36**
PCR_7	**AB 3D 75 4C 40 CF C0 78 99 8F A3 E1 A6 92 C6 01 67 92 F1 C5**

.
.
.

Up To **PCR_{23}** Now,

CoT_{d0} = { PCR_0

= 663F8EE75A5C33FDEE81AF20FA9E7F651F479135,
(298df125b260ef64201bdf0815c003873eedd50e [BIOS:EV_S_CRTM_VERSION],
B794d4c2eddf03b69bdc8ac0a1cbf9278fe131ca0 [BIOS:EV_POST_CODE(EV_CODE_NOCERT)]
.
.
(name and value pairs for PCR_0))

PCR_1 = BBC045C19F21F54F49B13895EC49A9ECBF15F81D,
PCR_2 = 0CE8061EBA5C532C62595DDB3075CAE8A8575361, … Up To PCR_{23} }

Similarly , we received following CoT for domain-U

CoT_{dU} = {PCR_0 = 89A05EBEF181D53653680BDB59C75DC1E06AB2AB,
PCR_1 = AB8BA78A38E836D77AAA595F2A27A57E87090928, PCR_2 =
D32F78E10FC41274DC447AE01576FD547F2E4E36, …, Up To PCR_{23}}

Figure 8: PCR values and its corresponding domain CoTs

References

[1] A4Cloud, *The Cloud Accountability Project*, July 3, 2016. (http://www.a4cloud.eu/)

[2] I. M. Abbadi, "Clouds trust anchors," in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'12)*, pp. 127–136, 2012.

[3] M. Almorsy, J. Grundy, and I. Müller, "An analysis of the cloud computing security problem," in *Proceedings of APSEC Cloud Workshop*, pp. 8–18, 2010.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, 2003.

[6] D. Catteddu, M. Felici, G. Hogben, A. Holcroft, et al., "Towards a model of accountability for cloud computing services," in *International Workshop on Trustworthiness, Accountability and Forensics in the Cloud (TAFC'13)*, pp. 1–10, 2013.

[7] D. Contractor and D. Patel, "Analyzing trustworthiness of virtual machines in data-intensive cloud computing," in *2014 Twelfth Annual International Conference on Privacy, Security and Trust (PST'04)*, pp. 403–406, July 2014.

[8] J. G. Dyer, M. Lindemann, R. Perez, R. Sailer, L. Van Doorn, and S. W. Smith, "Building the IBM 4758 secure coprocessor," *Computer Journal*, vol. 34, no. 10, pp. 57–66, 2001.

[9] IMA, *Integrity Measurement Architecture (Linux IMA)*, July 3, 2016. (http://sourceforge.net/apps/mediawiki/linux-ima/index.php)

[10] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*. O'Reilly Media, 2009.

[11] F. Mayer, D. Caplan, and K. MacMillan, *SELinux by Example: Using Security Enhanced Linux*. Pearson Education, 2006.

[12] P. Mell and T. Grance, "The NIST definition of cloud computing," *National Institute of Standards and Technology*, vol. 53, no. 6, pp. 50, 2009.

[13] A. Nakhimovsky, T. Myers, *Google, Amazon, and Beyond: Creating and Consuming Web Services*, Springer, 2004.

[14] D. Nunez, C. Fernandez-Gago, S. Pearson, and M. Felici, "A metamodel for measuring accountability attributes in the cloud," in *IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom)*, vol. 1, pp. 355–362, 2013.

[15] S. Pearson, "Towards accountability in the cloud," *IEEE Internet Computing*, vol. 15, no. 4, pp. 64–69, 2011.

[16] S. Pearson, V. Tountopoulos, D. Catteddu, et al., "Accountability for cloud and other future Internet services," in *IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom'12)*, pp. 629–632, 2012.

[17] R. Perez, R. Sailer, and L. van Doorn, "vTPM: Virtualizing the trusted platform module," in *Proceedings of the 15th Conference on USENIX Security Symposium*, pp. 305–320, 2006.

[18] V. Scarlata, C. Rozas, M. Wiseman, D. Grawrock, and C. Vishik, "TPM virtualization: Building a general framework," in *Trusted Computing*, pp. 43–56, 2008.

[19] J. Schiffman, T. Moyer, H. Vijayakumar, T. Jaeger, and P. McDaniel, "Seeding clouds with trust anchors," in *Proceedings of the 2010 ACM workshop on Cloud Computing Security Workshop*, pp. 43–46, 2010.

[20] J. Singh, "Cyber-attacks in cloud computing: A case study," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 78–87, 2014.

[21] S. B. Sitkin and N. L. Roth, "Explaining the limited effectiveness of legalistic remedies for trust/distrust," *Organization Science Journal*, vol. 4, no. 3, pp. 367–392, 1993.

[22] S. W. Smith, *Trusted Computing Platforms: Design and Applications*, vol. 2, Springer, 2005.

[23] I. Srivastava, *South Korea Cyber-attacked on Korean War Anniversary*, June 29, 2013. (http://timesofindia.indiatimes.com/world/rest-of-world/South-Korea-cyber-attacked-on-Korean-war-anniversary/articleshow/20833246.cms?referral=PM)

[24] Trend Micro, *Trend Micro Investigates June 25 Cyber Attacks in South Korea*, July 1, 2013. (http://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/124/trend-micro-investigates-june-25-cyber\-attacks-in-south-korea)

[25] Trusted Computing Group, *TCG Infrastructure Working Group Reference Manifest (RM) Schema Specification, Version 1.0*, Nov. 16, 2006. (https://www.trustedcomputinggroup.org/wp-content/uploads/IWG-Reference_Manifest_Schema_Specification_v1.pdf)

[26] Trusted Computing Group, *TCG Software Stack (TSS) Specification, Version 1.2*, July 3, 2016. (http://www.trustedcomputinggroup.org/tcg-software-stack-tss-specification/)

[27] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[28] J. Yao, S. Chen, C. Wang, D. Levy, and J. Zic, "Accountability as a service for the cloud," in *2010 IEEE International Conference on Services Computing (SCC'10)*, pp. 81–88, 2010.

[29] A. R. Yumerefendi and J. S. Chase, "Trust but verify: accountability for network services," in *Proceedings of the 11th Workshop on ACM SIGOPS European Workshop*, pp. 37, 2004.

[30] M. Zareapoor, P. Shamsolmoali, and M. A. Alam, "Establishing safe cloud: Ensuring data security and performance evaluation," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 88–99, 2014.

[31] L. J. Zhang and Q. Zhou, "CCOA: Cloud computing open architecture," in *IEEE International Conference on Web Services (ICWS'09)*, pp. 607–616, 2009.

Dipen Contractor Dipen contractor received his B.E degree in computer engineering in 2008 & M.E. degree in computer science and engineering in 2011. He is pursuing Ph.D. in computer engineering at National Institute of Technology Surat. His research interests include Cloud computing, Information security, Trust management, Remote attestation, and programming with Trusted Platform Module.

Dhiren Patel Dr.Dhiren Patel is a professor of computer engineering at National Institute of Technology Surat. He carries more than 20 years of experience in Academics, Research & Development of Secure ICT Infrastructure Design. His research interests cover Security and Encryption Systems, Cloud Computing and IoT, Identity and Access Management, e-Voting, Advanced Computer Architecture etc. Besides numerous journal and conference articles, Prof. Dhiren has authored a book "Information Security: Theory & Practice" published by Prentice Hall of India (PHI) in 2008. He is actively involved in Indo-UK, Indo-Norway, and Indo-Japan security research collaborations.

A Verifiable E-voting Scheme with Secret Sharing

Lifeng Yuan[1,2], Mingchu Li[1,2], Cheng Guo[1,2], Weitong Hu[3], and Zhihui Wang[1]
(Corresponding author: Cheng Guo)

School of Software Technology, Dalian University of Technology[1]
No.8 Road, Jinzhou District, Dalian 116620, China
Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province[2]
No.8 Road, Jinzhou District, Dalian, 116620, China
School of Computer and Technology, Hangzhou Dianzi University[3]
No.1 Street 2, Xiasha High Education Zone, Hangzhou, 310018, China
(Email: guocheng@dlut.edu.cn)
(Received Nov. 28, 2015; revised and accepted Apr. 9 & May 31, 2016)

Abstract

Traditional e-voting schemes use centralized and non-transparent count centers, which leads to people's distrust of the centers and doubt of the voting on impartiality and correctness. In this paper, we propose a distributed and verifiable e-voting scheme based on Mignotte's threshold secret sharing scheme, which effectively balances the conflict of interest between voters and count centers. Additionally, this scheme can also resist potential attacks from malicious participants, and satisfy special requirements of e-voting, especially for privacy and accountability, which contradict each other. Moreover, voters take on the major computation load in our scheme, which effectively reduces the computation burden of the vote counter.

Keywords: Accountability, e-voting, secret sharing, verifiability

1 Introduction

As we know, voting is important in democratic society. In paper voting, it is possible for tally clerks to obtain and even tamper with the contents of voters' ballots during printing and delivery phase, so voters may doubt the authenticity of the final result. Moreover, paper voting takes great cost to count votes in the voting process. To fix these problems, a multitude of e-voting schemes based on various cryptographic techniques are developed, which are more convenient for voters to vote at any time and place. Therefore, e-voting is widely used to replace paper voting.

In recent years, various security technologies (such as mix-net [4, 5, 8, 24, 25], blind signature [3, 6, 12, 15, 29], homomorphic encryption [7, 9, 11, 26, 28] and secret sharing [10, 11, 17, 33]) provide a solid foundation for the development of e-voting. Compared with paper voting, an e-voting scheme should be able to satisfy more requirements [15, 21, 36], such as privacy, verifiability, fairness and transparency. Since one requirement may conflict with another (for example, accountability and privacy), it is challenging to satisfy all of them.

In this paper, we propose an e-voting scheme based on Mignotte's secret sharing schemes [23] with the following advantages:

1) It can balance the conflict of interest between voters and central vote counter by mutual supervision. In some verifiable voting schemes, the voter can only verify whether his/her own vote is computed correctly, but any voter in our scheme can verify whole vote result without affecting the privacy of the scheme, so their trust in this scheme can increase greatly.

2) It improves computational efficiency. Schemes that use central entities to execute all computation tasks often make central entities overloaded. However, in our scheme, voters take the majority of computation tasks. Meanwhile, for a single voter, the computation burden is acceptable.

3) It resolves the conflict between accountability and privacy. In our scheme, no one can obtain legal voters' selections. But, in order to identify attackers, the third-party authority can recover the voter's selection with t or more voters' assistance. However, this is inevitable and understandable. Note that the third-party authority also cannot obtain any voter' selection unless t or more voters agree.

Our scheme uses Mignotte's threshold secret sharing technique, which is based on the Chinese Remainder Theorem. Also, our scheme uses some special sequences of integers, referred to as Mignotte sequences. For reader's convenience, we describe Mignotte sequences as follows.

Let $n \in \mathbb{Z}$, $2 \leq t \leq n$. A (t, n) Mignotte sequence is a sequence of pairwise co-prime positive integers $p_1 < p_2 < \cdots < p_n$ such that

$$\prod_{i=0}^{t-2} p_{n-i} < \prod_{i=1}^{t} p_i \tag{1}$$

Given a publicly known (t, n) Mignotte sequence, the scheme works as follows:

1) The dealer chooses a secret $s \in \mathbb{Z}$, such that

$$\prod_{i=0}^{t-2} p_{n-i} < s < \prod_{i=1}^{t} p_i. \tag{2}$$

2) For all $1 \leq i \leq n$, the secret share s_i for participant P_i is computed by the dealer as $s_i = s \bmod p_i$.

3) If the number of participants with secret shares is greater than or equal to t, the secret s can be recovered. Without loss of generality, assume that t participants $P_{i_1}, P_{i_2}, \cdots, P_{i_t}$ $(1 \leq i_1 < i_2 < \cdots < i_t \leq n)$ provide their secret shares $s_{i_1}, s_{i_2}, \cdots, s_{i_t}$, then the system of congruences can be built as

$$\begin{cases} s_{i_1} \equiv s \bmod p_{i_1} \\ s_{i_2} \equiv s \bmod p_{i_2} \\ \quad\quad \vdots \\ s_{i_t} \equiv s \bmod p_{i_t} \end{cases} \tag{3}$$

where $s_j \equiv s \bmod p_j$ $(j \in \{i_1, i_2, \cdots, i_t\})$ means $s_j \bmod p_j = s \bmod p_j$. Let $P = \prod_{k=1}^{t} p_{i_k}$. For all $1 \leq k \leq t$, $R_{i_k} = P/p_{i_k}$ and $R_{i_k} r_{i_k} \equiv 1 \bmod p_{i_k}$. Then, the secret s can be recovered as

$$s = \sum\nolimits_{k=1}^{t} s_{i_k} R_{i_k} r_{i_k} \bmod P. \tag{4}$$

The rest of this paper is organized as follows: we present the related work in Section 2, and introduce some concepts of e-voting in Section 3. Section 4 describes the details of our e-voting scheme. In Section 5, we discuss correctness, security, features, computation complexity of our scheme, and then, compare our scheme with related schemes. Finally, Section 6 concludes this paper.

2 Related Work

In recent years, various e-voting schemes have been developed [18, 19, 20], and these schemes are based on different security methods, such as mix-net [4, 5, 8, 24, 25], blind signature [3, 6, 12, 15, 29], homomorphic encryption [7, 9, 11, 26, 28] and secret sharing [10, 11, 17, 33]. However, to the best of our knowledge, existing e-voting schemes cannot satisfy all the special requirements, which will be discussed in Section 3.1.

In 1981, Chaum [5] proposed the first e-voting scheme based on public-key cryptosystem and mix-net. His scheme can be described as follows: firstly, each voter authenticates himself/herself and then sends his/her encrypted vote. The vote counter collects all votes and terminates the corresponding relationship between the ballots and voters by mix-net. Finally, these votes will be decrypted and counted. Note that this scheme requires a lot of computation to guarantee that each vote be properly processed, so it is inefficient and not suitable for large-scale voting. Moreover, because the mix-net is not transparent, voters may doubt the correctness of the vote result. More schemes based on mix-nets can be referred in [4, 8, 24, 25].

Blind signature, introduced by Chaum [3] in 1983, allows an authority to sign an encrypted message without knowing the plaintext. Fujioka et al. [15] proposed a FOO e-voting scheme, and then, Cranor and Cytron [12] implemented the FOO scheme named Sensus. The FOO scheme also has problems like ballot collusion. More schemes [6, 29] based on blind signature were proposed afterwards, for example, Radwin [29] proposed an untraceable, universally verifiable voting scheme and Chen [6] proposed a receipt-free voting scheme using double-trapdoor commitment.

In 1985, Cohen and Fischer [9] proposed a voting scheme based on homomorphic encryption. Exploiting the homomorphism of certain encryption algorithms, the schemes [7, 11, 26, 28] do not decrypt a single ballot, but decrypt the product of all ballots to get the vote result. They are efficient in yes/no voting, while in other types of voting, they have low efficiency. In these schemes, voters need to use zero-knowledge proofs to prove the correctness of their votes. If the voting is complex (such as selecting $|n/2 - 1|$ people from n candidates), these schemes will require lots of computation.

In 1979, Shamir [34] and Blakley [1] proposed the concept of secret sharing independently, and built a (t, n) threshold secret sharing scheme respectively. Subsequently, many researchers further studied secret sharing technology [13, 14, 22, 27]. The secret sharing scheme has also been applied to e-voting, for example, schemes in [10, 11, 33] based on Shamir's polynomial interpolation secret sharing scheme, and Iftene's scheme [17] based on Chinese remainder theory. Since these schemes use centralized entities without transparency, voters cannot verify the vote result and thus may doubt the correctness of the vote result. Moreover, these schemes lack consideration about the impact of attackers. In this paper, we propose a verifiable e-voting scheme based on Mignotte's threshold secret sharing scheme. Our scheme enable voters to verify the vote result, and balances the conflict of interest between voters and the central vote counter. Additionally, this scheme can also resist potential attacks from malicious participants.

In addition, other new methods, such as image hiding [31] and quantum mechanism [2], were also applied to e-voting schemes, and some e-voting schemes use multiple methods together. For example, Cohen and Fisher's scheme [9] uses mix-net and blind signature together.

3 Preliminaries

In this section, we will discuss the e-voting requirements, the classification of e-voting, the participants of our scheme and the attack models.

3.1 Requirements of E-voting Scheme

Some literatures [15, 16, 21, 32, 35, 36] have discussed the requirements of e-voting schemes. The most important requirements are as follows:

1) **Legality**: only legal participants can vote.
2) **Correctness**: if all participants are honest and execute the schemes strictly, the vote result can reflect voters' intentions correctly.
3) **Privacy**: no one can obtain another voter's personal vote information.
4) **Robustness**: attackers cannot disrupt the vote procedure.
5) **Verifiability**: each voter can independently check the correctness of the vote result.
6) **Fairness**: each voter only knows his/her own vote information, and cannot know the final vote result until the vote has been finished.
7) **Transparency**: the whole voting process and all technologies used in the voting scheme are transparent to each voter.
8) **Uniqueness**: a voter is not allowed to vote more than once.
9) **Accountability**: attackers can be revealed and punished.

Some requirements conflict with each other (such as privacy and accountability), so it is very challenging to satisfy all of them.

3.2 Classifications of E-voting

Up to now, many e-voting models are discussed, and e-voting models are classified into 5 types [30]:

1) **Yes/No voting**: every voter votes for or against the candidates.
2) **1-out-of-L voting**: every voter votes for one candidate from the set of L candidates.
3) **K-out-of-L voting**: every voter votes for K different candidates from the set of L candidates.
4) **K-sort-out-of-L voting**: every voter votes for K ordered different candidates from the set of L candidates. The order of the selected candidates represents the importance.
5) **1-L-K voting**: the voter picks out a subset of L candidates, and then chooses K candidates form this set.

Note that above five types of voting are relative. In fact, 1-out-of-L voting is the generalization of the other four types of voting. For example, taking $L = 2$, we obtain Yes/No voting. If 1-out-of-L voting executes K times, we build K-out-of-L voting and K-sort-out-of-L voting. The 1-L-K voting can be considered as the combination of 1-out-of-L voting and K-out-of-L voting. Hence, without loss of generality, we only consider 1-out-of-L voting in our schemes.

3.3 Participants

In our scheme, there are $n + 3$ ($n > 2$) participants, including n voters, a trusted third-party authority, a vote counter and a registration center. The obligations of these participants are as follows:

1) n voters: each voter casts his/her vote, and computes the authentication value. When voters doubt the vote result which is computed and broadcasted by the vote counter, they can verify it using the authentication value. If the verification is unsuccessful, voters can report to the third-party authority who can investigate malicious participants.
2) The third-party authority: the third-party authority is responsible for initializing the e-voting scheme and investigating attackers. In setup phase, he/she prepares for the voting, and selects suitable parameters to ensure the security of the scheme. When attackers are detected in the verification phase, the third-party authority will initiate the investigation and find out them.
3) The registration center: the registration center is responsible for registering all applicants who want to join the e-voting. He/she verifies each applicant's personal information, and then, assigns a unique identification code for the applicant who passes the verification.
4) The vote counter: the vote counter computes the final vote result in the vote tallying phase, and then, broadcasts it.

In our e-voting scheme, we assume that the third-party authority and registration center are trusted.

3.4 Attack Model

The activities of attackers will threaten the security and privacy of the e-voting scheme. They intend to obtain the vote selections of the legal voters and modify the final vote result.

There are two types of attacks. One type is the Single Attack, involving only one attacker. In our scheme, third-party authority and the registration center are trusted,

so we only discuss the vote counter and the voter's single attack. The other type is Collusion Attack, which has multi-attackers. Since the vote counter's behaviors will be verified in verification phase, there is no need to consider it as a collusion attacker. In this paper, we only discuss the collusion attack launched by voters. In Section 5, we will discuss all types of malicious participants' attacks in detail.

4 The Proposed Scheme

In our scheme, the participants include a third-party authority $A_{uthority}$, a registration center $R_{egistrar}$, a vote counter C_{ounter} and n voters $V_1, V_2, \cdots, V_n$. $A_{uthority}$ and $R_{egistrar}$ are trusted. As we mentioned in Section 3.2, without loss of generality, we use 1-out-of-L voting in our scheme. We also need the following notations and parameters in Table 1.

Table 1: Parameters declaration

Symbol	Meaning
n	the number of voters
L	the number of candidates
C_i	the candidate i
c_i	the candidate value representing C_i
vn_i	the number of voters who vote for C_i
Id_i	$V_i's$ identification code
V_i	the voter i
v_i	$V_i's$ vote selection, $v_i \in \{c_1, c_2, \cdots, c_L\}$
m_i	$V_i's$ vote mask
M	the sum of all masks, $M = \sum_{i=1}^{n} m_i$
p_i	the corresponding prime of V_i
B_i	$V_i's$ ballot
$B_{i,j}$	the sub-ballot which V_i sends to V_j
MST_i	the masked sub-tally computed by V_i
MT	the masked tally, $MT = \sum_{i=1}^{n} B_i$
T	the tally, $T = \sum_{i=1}^{n} v_i$

In the following, we will describe our e-voting scheme. Note that all related technologies and vote process are transparent to voters.

4.1 Setup

In setup phase, $A_{uthority}$ prepares for the voting, and selects suitable parameters to ensure the security of the scheme. He/she works as follows:

1) $A_{uthority}$ generates a candidate value sequence $c_1, c_2, \cdots, c_L \in \mathbb{Z}$ which satisfies

$$c_i > n \times c_{i+1} \quad (i = 1, 2, \cdots, L-1) \tag{5}$$

If a voter chooses c_i, it means that this voter votes for candidate C_i.

2) $A_{uthority}$ selects suitable threshold according to the security requirement of the scheme.

3) $A_{uthority}$ generates a (t, n) Mignotte sequence $p_1, p_2, \cdots, p_n$, and this sequence satisfies Equation (1).

4) $A_{uthority}$ generates an integer mask sequence $m_1, m_2, \cdots, m_n$, and computes the sum of all masks $M = \sum_{i=1}^{n} m_i$.

In order to ensure the security and privacy of the scheme, the above parameters should satisfy the following conditions.

Condition 1. *In order to ensure attackers cannot reduce the guess scope of vote's selection v_i even if attackers obtain the voter's ballot B_i, the candidate value sequence and the mask sequence need to satisfy*

$$c_L + \min\{m_1, m_2, \cdots, m_n\} > c_1. \tag{6}$$

Condition 2. *In following phase, we need to recover masked tally MT and each voter's ballot B_i $(1 \le i \le n)$, so MT and B_i need to satisfy $\prod_{k=0}^{t-2} p_{n-i} < B_i, MT < \prod_{k=1}^{t} p_i$. Since $MT = \sum_{i=1}^{n} B_i$ and $B_i = v_i + m_i$, the candidate value sequence and the mask sequence need to satisfy*

$$\begin{cases} n \times c_1 + M < \prod_{i=1}^{t} p_i \\ \prod_{i=0}^{t-2} p_{n-i} < c_L + \min\{m_1, m_2, \cdots, m_n\} \end{cases} \tag{7}$$

4.2 Registration

Each voter sends his/her personal information to $R_{egistrar}$. If the applicant's personal information is legitimate, $R_{egistrar}$ assigns a unique identification code to her/him. By the signature technology which is represented in Section 4.3, participants can know the corresponding relation between the sender and his/her information.

According to voters' registration information, $A_{uthority}$ sends mask m_i and prime p_i to voter V_i through secure communication channels for all $1 \le i \le n$. Then, $A_{uthority}$ broadcasts the sum of all masks M, candidate and candidate value pairs $< c_i, C_i >$ $(1 \le i \le L)$, and voter's identification code and corresponding prime pairs $< id_i, p_i >$ $(1 \le i \le n)$.

4.3 Signature

In our scheme, the data, which $A_{uthority}$ use to identify attackers in investigation phase, may be stored by malicious participants. Thus, we use signature technique to ensure the reliability of the data and their source. In this way, senders cannot deny the data they sent, and $A_{uthority}$ can

easily detect the malicious data provider who tamper the data.

Some signature technologies can be used in our scheme to create the corresponding relations between the sender and his/her information. For example, we can use RSA signature technology which works as follows: each participant will generate he/her private key and public key and send public key to $R_{egistrar}$. After getting these public keys, $R_{egistrar}$ will broadcast the corresponding relation between participants' identification codes and their public key. If a participant wants to send a message, he/she will encrypt this message and his/her identification code using private key. After receiving the message, the receiver can decrypt the message using the sender's public key and verify the identity of the sender. In this way, the corresponding relation between the sender and his/her information can be built up. Here, we will not describe the details of RSA signature technology.

4.4 Vote

Following voter $V_i's$ own will, he/she chooses a vote selection $v_i \in \{c_1, c_2, \cdots, c_L\}$ and forms ballot $B_i = v_i + m_i$. For example, if V_i wants to vote for candidate C_k $(1 \leq k \leq L)$, he/she can choose $v_i = c_k$. Then, V_i computes each sub-ballot $B_{i,j} = B_i \bmod p_j$ $(1 \leq j \leq n, j \neq i)$, and send it to corresponding voter V_j.

After receiving all sub-ballot $B_{j,i}$ from voter V_j $(1 \leq j \leq n$ and $j \neq i)$, voter V_i checks the number of each $B_{j,i}$. If a voter sends his/her sub-ballot more than once and his/her sub-ballots are different, voter V_i will report to $A_{uthority}$ and use the last $B_{j,i}$. Then, voter V_i computes the masked sub-tally

$$MST_i = \sum_{j=1}^{n} B_{j,i} \bmod p_i, \tag{8}$$

and then, sends it to C_{ounter}.

4.5 Vote Tallying

In this phase, C_{ounter} randomly selects t voters $V_{i_1}, V_{i_2}, \cdots, V_{i_t}$ $(1 \leq i_1 < i_2 < \cdots < i_t \leq n)$, and then, uses their masked sub-tallies $MST_{i_1}, MST_{i_2}, \cdots, MST_{i_t}$ to build the system of congruences, such that

$$\begin{cases} MST_{i_1} \equiv MT \bmod p_{i_1} \\ MST_{i_2} \equiv MT \bmod p_{i_2} \\ \qquad \vdots \\ MST_{i_t} \equiv MT \bmod p_{i_t} \end{cases} \tag{9}$$

where $MT = \sum_{i=1}^{n} B_i$ and, for all $j \in \{i_1, i_2, \cdots, i_t\}$,

$$MST_j = \sum_{i=1}^{n} (B_i \bmod p_j) \bmod p_j. \tag{10}$$

C_{ounter} computes masked tally MT, using the general variant of the Chinese remainder theorem (see details in Section 1). Then, tally T, which can be described as $T = \sum_{i=1}^{L} (vn_i \times c_i)$, can be computed as $T = \sum_{i=1}^{n} v_i = \sum_{i=1}^{n} (B_i - m_i) = MT - M$. Let $r_0 = T$, the vn_i, which presents the number of the voters who vote for candidate C_i, can be computed as

$$\begin{cases} vn_i = \left\lfloor \frac{r_{i-1}}{c_i} \right\rfloor \\ r_i = r_{i-1} \bmod c_i \end{cases} (i = 1, 2, \cdots, L). \tag{11}$$

After recovering the final vote result, C_{ounter} broadcasts the masked tally MT and the final vote result $< vn_i, C_i >$ $(1 \leq i \leq L)$.

4.6 Verification and Investigation

In this section, we introduce our verification and investigation methods.

Verification A: If voter V_i doubt the vote result, he/she can verify masked tally MT which is computed and broadcasted by C_{ounter} as

$$MST_i = MT \bmod p_i. \tag{12}$$

If $MST_i \neq MT \bmod p_i$, verifier V_i will report to $A_{uthority}$ who can investigate C_{ounter}'s misbehavior. The investigation steps are described as follows:

1) Using Equation (12), $A_{uthority}$ verifies all masked sub-tallies $MST_1, MST_2, \cdots, MST_n$. If the number of voters whose masked sub-tallies satisfy Equation (12) is less than t, $A_{uthority}$ will know the masked tally MT was forged by C_{ounter}, and then, he/she broadcast C_{ounter}'s misbehaviors and finish the investigation.
2) $A_{uthority}$ selects t voters $V_{j_1}, V_{j_2}, \cdots, V_{j_t}$ whose masked sub-tallies satisfy Equation (12), and gets all voters' sub-ballots from these voters. Then, all voters' ballots can be recovered. For example, voter $V_i's$ ballot B_i can be recovered, using the general variant of the Chinese Remainder Theorem. The system of congruences can be built as

$$\begin{cases} B_{i,j_1} \equiv B_i \bmod p_{j_1} \\ B_{i,j_2} \equiv B_i \bmod p_{j_2} \\ \qquad \vdots \\ B_{i,j_t} \equiv B_i \bmod p_{j_t} \end{cases} \tag{13}$$

3) $A_{uthority}$ computes vote selection $v_i = B_i - m_i$, for all $1 \leq i \leq n$, and then he/she can find all attackers by recovering all phases of the e-voting (see details in Section 5.1).

Verification B: After verifying masked tally MT, voter V_i can verify vn_i, the number of the voters who vote for candidate C_i, using Equation (11). If the number is different from the number broadcasted by C_{ounter}, voter will report to $A_{uthority}$. Then $A_{uthority}$ will check the vote result. If C_{ounter} forged the vote result, $A_{uthority}$ will detect it.

5 Discussions

5.1 Correctness Analysis

In this section, we discuss the correctness of our scheme. By Proposition (1), we know that if all voters follow the vote rules, C_{ounter} can count the right vote result which reflects voters' true will.

Proposition 1. *If all voters follow the vote rules, C_{ounter} can count the right vote result which reflects voters' true will.*

Proof. Without loss of generality, in vote tallying phase, assume that C_{ounter} selects t voters $V_1, V_2, \cdots, V_t$ and uses their masked sub-tallies $MST_1, MST_2, \cdots, MST_t$ to recover masked tally MT. Since, for all $1 \leq i \leq t$,

$$\begin{aligned} MST_i &= \sum_{j=1}^{n} B_{j,i} \bmod p_i \\ &= \sum_{j=1}^{n} (B_j \bmod p_i) \bmod p_i \\ &= \sum_{j=1}^{n} B_j \bmod p_i \\ &= MT \bmod p_i \end{aligned} \tag{14}$$

C_{ounter} can build the system of congruences

$$\begin{cases} MST_1 \equiv MT \bmod p_1 \\ MST_2 \equiv MT \bmod p_2 \\ \quad\vdots \\ MST_t \equiv MT \bmod p_t \end{cases} \tag{15}$$

According to Condition 2, i.e., $\prod_{i=0}^{t-2} p_{n-i} < MT < \prod_{i=1}^{t} p_i$, C_{ounter} can recover MT by the general variant of the Chinese remainder theorem.

Tally T is the sum of all voters' selection v_i $(i = 1, 2, \cdots, n)$, so it can be computed as

$$T = \sum_{i=1}^{n} v_i = \sum_{i=1}^{n} (B_i - m_i) = MT - M \tag{16}$$

In addition, T can be described as

$$T = \sum_{j=1}^{n} v_j = \sum_{i=1}^{L} (vn_i \times c_i), \tag{17}$$

where $v_j \in \{c_1, c_2, \cdots, c_L\}$ and $\sum_{i=1}^{L} vn_i = n$. For all $1 \leq i \leq L-1$, we know that $c_i > n \times c_{i+1}$ and $0 \leq vn_i < n$, such that

$$c_i > \sum_{j=i+1}^{L} (vn_j \times c_j). \tag{18}$$

Thus, for all $1 \leq i \leq L$, if we set

$$\begin{cases} r_0 = T = \sum_{j=1}^{L} (vn_j \times c_j) \\ r_i = r_{i-1} \bmod c_i = \sum_{j=i+1}^{L} (vn_j \times c_j) \end{cases} \tag{19}$$

vn_i can be computed as follows:

$$\begin{aligned} \left\lfloor \frac{r_{i-1}}{c_i} \right\rfloor &= \left\lfloor \frac{\sum_{j=i}^{L} (vn_j \times c_j)}{c_i} \right\rfloor \\ &= vn_i + \left\lfloor \frac{\sum_{j=i+1}^{L} (vn_j \times c_j)}{c_i} \right\rfloor \\ &= vn_i. \end{aligned} \tag{20}$$

Obviously, in our scheme, if all voters follow the vote rules, C_{ounter} can compute the right vote result which reflects voters' true will. □

5.2 Security Analysis

In the e-voting scheme, malicious participants may attack the system. Here, we will analyze this problem in detail according to the attack models mentioned above.

5.2.1 Single Attack

According to participants' property, there are two types of single attacker, i.e., C_{ounter} and voter. We will analyze their misbehavior as follows:

1) Malicious C_{ounter}.

   C_{ounter} engages in two kinds of misbehavior: one is vote result cheating, the other is privacy stealing.

   Vote result cheating: if C_{ounter} forges masked tally and the final vote result to cheat voters, it will be discovered by voters in the verification phase.

   Privacy stealing: even if C_{ounter} wants to acquire voter $V_i's$ $(1 \leq i \leq n)$ vote selection v_i by collecting related information, he/she cannot recover voter $V_i's$ ballot B_i from prime number p_i, masked sub-tally MST_i and masked tally MT.

2) Malicious voter.

   Assume that voter V_i is malicious voter, whose attacks can be involved in the following attack cases:

   Case A. Using illegal ballots IB_i;

   Case B. Voting more than once;

   Case C. Sending different sub-ballots to other voters;

   Case D. Sending illegal masked sub-tally $IMST_i$ to C_{ounter};

   Case E. Trying to obtain legal voters' vote selections (such as voter $V_j's$ vote selection v_j).

   Case A, B, C and D can influence correctness and Case E can influence privacy. The security analysis of single malicious voter's attack is as follows:

   Case A. In the investigation phase, $A_{uthority}$ can recover voter $V_i's$ ballot B_i by solving the system of Congruences (13). Then, vote selection v_i is computed as $v_i = B_i - m_i$. If

$v_i \notin \{c_1, c_2, \cdots, c_L\}$, voter $V_i's$ misbehavior will be detected.

Case B. $R_{egistrar}$ will assign a unique identification code Id_i for voter V_i. According to Section 4.2, participants know the corresponding relation between the sender and his/her information. If voter V_i vote a new ballot $B_{i_{\text{new}}}$ and $B_{i_{\text{new}}} \neq B_i$, the information receivers will detect this misbehavior.

Case C. If voter V_i sends different sub-ballots to other voters, $A_{uthority}$ will recover an illegal ballot IB_i in the investigation phase by solving the system of Congruences (13). Thus, this misbehavior will be detected as in Case A..

Case D. According to the definition of threshold secret sharing, C_{ounter} just needs t honest voters to recover the masked tally MT. If only voter V_i sends an illegal masked sub-tally $IMST_i$, C_{ounter} can still recover MT. Moreover, $A_{uthority}$ can check $IMST_i$ in the investigation phase using Equation (12). If $IMST_i \neq MT \bmod p_i$, $A_{uthority}$ will detect voter $V_i's$ attack.

Case E. Voter V_i only gets voter $V_j's$ sub-ballot $B_{j,i}$, which is computed as $B_{j,i} = (v_j + m_j) \bmod p_i$, so he/she cannot compute ballot B_j. Even Voter V_i obtain ballot B_j, he/she also cannot recover voter $V_j's$ vote selection v_j without mask m_j which only $A_{uthority}$ and voter V_j know.

The above analysis shows that the attacks, launched by a single malicious participant, can be resisted in our e-voting scheme.

5.2.2 Collusion Attack

Since C_{ounter}'s behaviors will be verified in verification phase, there is no need to consider it as a collusion attacker. In this paper, we only discuss the collusion attack launched by voters.

Collusion voters' attacks can be involved in the following attack cases: A. Modifying the vote result; B. Obtaining legal voters' vote selection (such as voter $V_i's$ vote selection v_i). We analyze the above two cases next.

Case A: In the setup phase, $A_{uthority}$ selects proper threshold t. Generally, we set $t \geqslant \lceil (n+1)/2 \rceil$. If the number of collusion voters is more than or equals to t, they can win the voting and this collusion attack is meaningless. On the other hand, if the number of collusion voters is less than t, they cannot forge enough masked sub-tallies to cheat C_{ounter}. C_{ounter} will recover illegal masked tally IMT which cannot pass the verification phase. Then this collusion attack will be detected by $A_{uthority}$ in the investigation phase.

Case B: If the number of collusion voters is less than t, legal voter $V_i's$ ballot B_i will not be recovered. Otherwise, there will be a situation in which t voters win the voting and they still want to know voter $V_i's$ vote selection v_i. In this situation, ballot B_i can be recovered by solving the system of Congruences (13). Since $B_i = v_i + m_i$, vote selection v_i still cannot be computed without mask m_i which is only known to $A_{uthority}$ and voter V_i. Moreover, according to Condition 1, they also cannot reduce the guess scope of vote selection v_i, even if ballot B_i was obtained (see Section 4.1).

5.3 Features Analysis

In this section, we will analyze the features of our scheme according to the requirements in Section 3.1.

Legality: $R_{egistrar}$ verifies voters' personal information and assigns a unique identification code to each legal voter, so illegal voters cannot be involved in our scheme.

Correctness: If each voter is honest and strictly executes our scheme, C_{ounter} can recover the correct masked tally MT by solving the system of Congruences (9). Then, C_{ounter} can compute the final vote result that reflects voters' true will, using Equation (11).

Privacy: Protecting the privacy of voters' selections is one of the most important security requirements. By using the threshold secret sharing technology and the mask codes, the content of vote selection in our scheme is hidden to ensure privacy. According to the analysis in Section 5.2, any person cannot obtain voter' selection without the corresponding mask which is only known to this voter and $A_{uthority}$, and $A_{uthority}$ cannot recover any voter' selection unless t or more voters agree either. Obviously, our scheme can protect the privacy of voters' selections very well.

Robustness: From the analysis in Section 5.2, we know our scheme can resist all attacks launched by voters and the counter.

Verifiability: In the verification phase, voters can verify masked tally MT and the final vote result, with their own information (see details in Section 4.6).

Fairness: In the process of voting, each voter only gets other voters' sub-ballots (for example, voter V_j gets voter $V_i's$ sub-ballot $B_{i,j}$ which is computed as $B_{i,j} = B_i \bmod p_j$), so he/she cannot recover the final vote result using these information until C_{ounter} broadcasts it.

Uniqueness: In Section 4.3, the corresponding relation between voters' information and their identification code is established. If the voter casts his/her vote more than once, it can be detected in the vote phase.

Table 2: Computation complexities of all phases

Phases	Voter	Counter	Authority	Registrar
Setup	-	-	$O(n)$	-
Registration	-	-	-	$O(n)$
Vote	$O(n)$	-	-	-
Vote tallying	-	$O(n)$	-	-
Verification	$O(1)$	-	-	-
C_{ounter} *investigation*	-	-	$O(n)$	-
Voter investigation	-	-	$O(n)$	-
Sum	$O(n)$	$O(n)$	$O(n)$	$O(n)$

Transparency: In our scheme, we need participants to supervise their behaviors mutually, so that all working mechanisms and voting process are transparent to all participants in the whole process.

Accountability: According to the security analysis in Section 5.2, $A_{uthority}$ can find attackers and then reduce the damage. At the same time, $A_{uthority}$'s existence can also deter some attackers.

From the above analysis, our scheme obviously satisfies all the requirements and balances the conflict between privacy and accountability.

5.4 Computation Complexity Analysis

In order to analyze the computation cost of our scheme more clearly, we analyze the computation cost of voting and the computation cost of signature in this section, independently.

5.4.1 Computation Cost Analysis of Voting

We list the computation complexities of all work phases in Table 2.

In setup phase, $A_{uthority}$ should prepare for e-voting and select suitable parameters. $A_{uthority}$'s computation complexity is $O(n)$.

In registration phase, $R_{egistrar}$ should confirm whether the information of each applicant is legitimate, and then, assign a unique identification code to the legal applicant. In this phase, $R_{egistrar}$'s computation complexity is $O(n)$.

In vote phase, voters need to form the ballot and compute the masked sub-tally. Each voter's computation complexity is $O(n)$, so the computation complexity of n voters is $O(n^2)$.

In vote count phase, C_{ounter} need to recover the masked tally MT and compute the vote result. The computation complexity of recovering MT which need to solve the system of t congruence equations is $O(n)$, and the computation complexity of computing the vote result is $O(L)$ $(L < n)$, so the computation complexity of vote tallying phase is $O(n)$.

In verification phase, if the voter doubt the vote counter, he/she can verify the masked tally and the vote result. The verifier's computation complexity is $O(1)$.

In investigation phase, the computation complexity of investigating the vote counter is $O(n)$, and the computation complexity of investigating a voter, by recovering this voter's ballot, is $O(n)$. In worst situation, $A_{uthority}$ need to investigate n voters and the computation complexity is $O(n^2)$. In reality, the majority of participants are honest and follow the rules of the vote scheme, so $A_{uthority}$ only need to investigate a small number of voters.

Since the computation of masked sub-tally which has the highest computation complexity is allocated to n voters, each voter's computation complexity is $O(n)$. C_{ounter} and $A_{uthority}$'s computation complexity also is $O(n)$. In our scheme, each participant's computation load is balanced, which effectively avoid overload of the vote counter.

5.4.2 Computation Cost Analysis of Signature

In our scheme, voter's sub-ballot is stored by other voters. When $A_{uthority}$ needs to recover a certain voter's vote selection, he or she needs not less than t voters to provide their stored information about this voter. Since these information providers may be malicious, we need to guarantee information is reliable and not tampered. In our scheme, we use signature technology to guarantee information reliability and validity. Table 3 lists all participants' computation cost of signature and verification, which also reflect the communication situation between participants.

Since multiple signature techniques can be used in our scheme, we use T_s to represent the computation cost of signing a message, use T_v to represent the computation cost of verifying a message, and use T to represent the computation cost of signing and verifying a message. By Table 3, we know the total computation complexity of signature is $O(n^2T)$. The major computation cost generated by signature is in vote phase, because every voter needs to send her/his sub-ballot to other $n-1$ voters. In addition, for a single participant, signature computation complexity is $O(nT)$, which is acceptable.

In addition, we can avoid the computation cost of signature through requiring voters send backup information to trusted entity. For example, in our scheme, voters can send information to designated trusted entity (e.g., $A_{uthority}$). Since the trusted entity can ensure the data and their sources are valid and correct, $A_{uthority}$ can use them to identify attackers. Note that, when trusted entity receives the backup information, he/she should check data consistency with the information receiver.

5.5 Comparisons

In this section, we compare the functionality and computation complexity with related schemes.

5.5.1 Functionality Comparisons

Table 4 compares our scheme's functionality with Cramer et al.'s scheme [10], Iftene's scheme [17] and Li et al.'s scheme [21]. We explain Table 4 as follows:

1) All four schemes have verification function. In Cramer et al.'s and Iftene's schemes, multiple counters compute vote result, and then, implement mutual verification in order to verify the correctness of the vote result. In Li et al.'s scheme, after vote result is announced, the voter can verify her/his own vote, but cannot verify the whole vote result. Thus, in their schemes, voters may doubt the correctness of the vote result. In our scheme, any voter can verify whole vote result, which greatly improve vote result's trustworthiness.

2) Iftene's scheme lack consideration of the impact of attackers, so it cannot resist attacks. Cramer et al.'s and Li et al.'s schemes can only resist the attacks launched by voters. Our scheme can resist the attacks launched by voters and the counter. Thus, our scheme is more secure and feasible.

3) All four schemes can protect the privacy of voters' vote selection, and only Li et al.'s and our schemes can identify attackers by recovering voters' selections. But, in our scheme, $A_{uthority}$ cannot obtain any voter's selection unless not less than t voters agree, which can better balance the privacy and the accountability.

4) Compared with three other schemes, our scheme can balance the participants' computation overload, which can effectively avoid overloading the central counter.

5.5.2 Computation Complexity Comparisons

Table 5 compares computation complexity of our scheme with Cramer et al.'s scheme [10], Iftene's scheme [17] and Li et al.'s scheme [21].

Our scheme and Li et al.'s use some techniques (e.g., signature technique) to ensure the reliability of the data and their source, which authority use to identify attackers. Thus, our scheme and Li et al.'s have higher computation cost than Cramer et al.'s and Iftene's schemes. But, in order to ensure the robustness and accountability of the e-voting scheme, it's inevitable. In addition, our scheme distributes the computation burden to all of participants, thus improve computational efficiency, and avoid overloading the counter.

6 Conclusions

In this paper, we propose an e-voting scheme which allows voters to verify the final vote result independently and balances the conflict of interest between voters and the vote counter. Moreover, the scheme is secure because it can resist attacks effectively. In this scheme, we suppose that the third-party authority and the registration center are credible. However, they might be non-credible in reality. Therefore, we plan to design a new mechanism which can avoid the supposition of those credible participants in the future.

Acknowledgments

A 5-page preliminary version of this paper appeared in the International Conference on Communication Technology, pp. 304-308, October, 2015. In addition, this paper is supported by the Nature Science Foundation of China under grant No. 61272173, 91315302, 61401060, 61501080, 61572095, the general program of Liaoning Provincial Department of Education Science Research under grants L2014017, and the Fundamental Research Funds for the Central Universities under grant No. DUT16QY09.

References

[1] G. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the National Computer Conference*, pp. 313–317, Montvale: NCC, 1979.

[2] M. Bonanome, V. Buzek, M. Hillery, , and M. Ziman, "Toward protocols for quantum-ensured privacy and secure voting," *Physical Review A*, vol. 84, no. 2, pp. 022331, 2011.

[3] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*, pp. 199–203, Santa Barbara, CA, USA, 1983.

[4] D. Chaum, P. A. Ryan, and S. Schneider, "A practical voter-verifiable election scheme," in *Computer Security (ESORICS'05)*, pp. 118–139, Milan, Italy, 2005.

[5] D. L. Chaum, "Untraceable electronic mail, return add-resses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[6] X. Chen, Q. Wu, F. Zhang, and H. Tian et al., "New receipt-free voting scheme using double-trapdoor commitment," *Information Sciences*, vol. 181, no. 2, pp. 1493–1502, 2011.

Table 3: Computation costs of signature

Phases	**Voter**	**Counter**	**Authority**	**Registrar**	**Sum**
Setup	T_v	-	nT_s	-	nT
Registration	T	-	-	nT	$2nT$
Vote	$T_s + (n-1)T$	nT_v	-	-	n^2T
Vote tallying	T_v	-	nT_s	-	nT
Verification	-	-	-	-	-
C_{ounter} *investigation*	T_s	-	nT_v	-	nT
Voter investigation	T_s	-	tT_v	-	tT
Sum	$(n+2)T + T_s$	nT_v	$n(T+T_s) + tT_v$	nT	$(n^2+5n+t)T$

[1] T_s: The computation cost of signing a message
[2] T_v: The computation cost of verifying a message
[3] T: The computation cost of signing and verifying a message

Table 4: Functionality comparisons between our scheme and related schemes

Functionalities	**Cramer et al. [10]**	**Iftene [17]**	**Li et al. [21]**	**Our scheme**
Multiple counter	Yes	Yes	Yes	No
Trusted counter	Yes	Yes	Yes	No
Voting type	Yes/No	Yes/No	All	All
Verification function	Yes	Yes	Yes	Yes
Verifying vote result	Yes	Yes	No	Yes
Verifying own vote	No	No	Yes	No
Verifying by counter	Yes	Yes	No	No
Verifying by voter	No	No	Yes	Yes
Protecting privacy	Yes	Yes	Yes	Yes
Resisting attack	No	No	Yes	Yes
Identifying attacker	No	No	Yes	Yes
Robustness	No	No	Yes	Yes
Calculated balance	No	No	No	Yes

Table 5: Computation complexity comparisons between our scheme and related schemes

Phases	**Cramer et al. [10]**	**Iftene [17]**	**Li et al. [21]**	**Our scheme**
Setup	Null	$O(n)$	$O(n^2D)$	$O(nT)$
Registration	Null	Null	Null	$O(nT)$
Vote	$O(n^2\log^2 n)$	$O(n^2)$	$O(n^3+n^2D)$	$O(n^2T)$
Vote tallying	$O(n^2\log^2 n)$	$O(n)$	$O(n^2D)$	$O(nT)$
Verification	$O(n)$	$O(n)$	$O(n)$	$O(n)$
C_{ounter} *investigation*	Null	Null	Null	$O(nT)$
Voter investigation	Null	Null	$O(n^2+nD)$	$O(nT)$
Sum	$O(n^2\log^2 n)$	$O(n^2)$	$O(n^3+n^2D)$	$O(n^2T)$

[1] Null: don't have this phase
[2] T: the computation cost of message signing or verifying
[3] D: the computation cost of message backup or verification

[7] J. Clark and U. Hengartner, "Selections: Internet voting with over-the-shoulder coercion-resistance," in *Financial Cryptography and Data Security*, pp. 47–61, Gros Islet, ST Lucia, 2012.

[8] M. R. Clarkson, S. Chong, and A. C. Myers, *Civitas: A Secure Voting System*, Technical Report, Cornell University, TR 2007-2081, May 2007.

[9] J. D. Cohen and M. J. Fischer, *A Robust and Verifiable Cryptographically Secure Election Scheme*, Technical Report, Yale University, YALEU/DCS/TR-416, July 1985.

[10] R. Cramer, M. Franklin, B. Schoenmakers, and M. Yung, "Multi-authority secret-ballot elections with linear work," in *International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'96)*, pp. 72–83, Saragossa, Spain, 1996.

[11] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," *European Transactions on Telecommunications*, vol. 8, no. 5, pp. 481–490, 1997.

[12] L. F. Cranor and R. K. Cytron, "Sensus: a security-conscious electronic polling system for the internet," in *System Sciences*, pp. 561–570, Wailea, HI, 1997.

[13] X. Dong, "A multi-secret sharing scheme based on the CRT and RSA," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 47–51, 2015.

[14] B. Feng, C. Guo, M. Li, and Z. Wang, "A novel proactive multi-secret sharing scheme," *International Journal of Network Security*, vol. 17, no. 2, pp. 123–128, 2015.

[15] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *Advances in Cryptology (AUSCRYPT'92)*, pp. 244–251, Gold Coast, Qld., Australia, 1993.

[16] D. A. Gritzalis, *Secure Electronic Voting*, USA: Springer US, 2003.

[17] S. Iftene, "General secret sharing based on the chinese remainder theorem with applications in e-voting," *Electronic Notes in Theoretical Computer Science*, vol. 186, no. 1, pp. 67–84, Springer, 2007.

[18] C. C. Lee, T. Y. Chen, S. C. Lin, and M. S. Hwang, "A new proxy electronic voting scheme based on proxy signatures," in *Lecture Notes in Electrical Engineering*, pp. 3–12, Dordrecht, Netherlands, 2012.

[19] C. T. Li and M. S. Hwang, "Security enhancement of chang-lee anonymous e-voting scheme," *International Journal of Smart Home*, vol. 6, no. 2, pp. 45–52, 2012.

[20] C. T. Li and M. S. Hwang, "A secure and anonymous electronic voting scheme based on key exchange protocol," *International Journal of Security and Its Applications*, vol. 7, no. 1, pp. 59–70, 2013.

[21] H. Li, Y. Sui, W. Peng, X. Zou, and F. Li, "A viewable e-voting scheme for environments with conflict of interest," in *2013 IEEE Conference on Communications and Network Security (CNS'13)*, pp. 251–259, Washington, DC, 2013.

[22] Y. J. Liu and C. C. Chang, "An integratable verifiable secret sharing mechanism," *International Journal of Network Security*, vol. 18, no. 4, pp. 617–624, 2016.

[23] M. Mignotte, "How to share a secret," in *Proceedings of the Workshop on Crytography*, pp. 371–375, Burg Feuerstein, Germany, 1983.

[24] C. Park, K. Itoh, and K. Kurosawa, "Efficient anonymous channel and all/nothing election scheme," in *Advances in Cryptology (EUROCRYPT'93)*, pp. 248–259, Lofthus, Norway, 1994.

[25] K. Peng, "A general and efficient countermeasure to relation attacks in mix-based e-voting," *International Journal of Information Security*, vol. 10, no. 1, pp. 49–60, 2011.

[26] K. Peng and F. Bao, "Efficient vote validity check in homomorphic electronic voting," in *11th International Conference on Information Security and Cryptology*, pp. 202–217, Seoul, South Korea, 2008.

[27] Q. Peng and Y. L. Tian, "A publicly verifiable secret sharing scheme based on multilinear diffie-hellman assumption," *International Journal of Network Security*, vol. 18, no. 6, pp. 1192–1200, 2016.

[28] A. A. Philip, S. A. Simon, and A. Oluremi, "A receipt-free multi-authority e-voting system," *International Journal of Computer Applications*, vol. 30, no. 6, pp. 15–23, 2011.

[29] M. J. Radwin, "An untraceable, universally verifiable voting scheme," in *Seminar in Cryptology*, pp. 829–834, Nagercoil, India, 1995.

[30] Z. Rjaskova, "Electronic voting schemes," in *Diplomova Praca*, pp. 70–76, Bratislava, Slovakia, 2002.

[31] L. Rura, B. Issac, and M. K. Haldar, "Secure electronic voting system based on image steganography," in *2011 IEEE Conference on Open Systems (ICOS'11)*, pp. 80–85, Langkawi, Malaysia, 2011.

[32] K. Sampigethaya and R. Poovendran, "A framework and taxonomy for comparison of electronic voting schemes," *Computers and Security*, vol. 25, no. 2, pp. 137–153, 2006.

[33] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," in *Advances in Cryptology (CRYPTO'99)*, pp. 148–164, Santa Barbara, CA, USA, 1999.

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

[35] F. Shirazi, S. Neumann, I. Ciolacu, and M. Volkamer, "Robust electronic voting: Introducing robustness in civitas," in *2011 International Workshop on Requirements Engineering for Electronic Voting Systems*, pp. 47–55, Trento, Italy, 2011.

[36] C. Staff, "Seven principles for secure e-voting," *Communications of the ACM*, vol. 52, no. 2, pp. 8–9, 2009.

Lifeng Yuan received the B.S. degree in Computer Science and Technology from Ningbo University in 2006 and the M.S. degree in software engineering from Dalian University of Technology in 2009. He is currently a Ph.D candiadate in Dalian University of Technology. His

current research interests include Secret Sharing, Data Hiding, Network and Information Security, and Image Processing.

Mingchu Li received the B.S. degree in mathematics, Jiangxi Normal University and the M.S. degree in applied science, University of Science and Technology Beijing in 1983 and 1989, respectively. He worked for University of Science and Technology Beijing in the capacity of associate professor from 1989 to 1994. He received his doctorate in Mathematics, University of Toronto in 1997. He was engaged in research and development on information security at Longview Solution Inc, Compuware Inc. from 1997 to 2002. From 2002, he worked for School of Software of Tianjin University as a full professor, and from 2004 to now, he worked for School of Software Technology of Dalian University of Technology as a full Professor, Ph.D. supervisor, and vice dean. His main research interests include theoretical computer science and cryptography.

Cheng Guo received the B.S. degree in computer science from Xi'an University of Architecture and Technology in 2002. He received the M.S. degree in 2006 and his Ph.D in computer application and technology, in 2009, both from the Dalian University of Technology, Dalian, China. From July 2010 to July 2012, he was a post doc in the Department of Computer Science at the National Tsing Hua University, Hsinchu, Taiwan. Since 2013, he has been an associate professor in the School of Software Technology at the Dalian University of Technology. His current research interests include information security and cryptology.

Wei-Tong Hu received the B.S. degree in 2008 and Ph.D. degree in 2015, both from the School of Software Technology, Dalian University of Technology, Dalian, China. Since 2016, he has been a lecturer in the School of Computer Science and Technology at Hangzhou Dianzi University. His current research interests include Secret Sharing, Data Hiding, Network and Information Security, and Image Processing.

Zhi-Hui Wang was born in Inner Mongolia in 1982. She received her B.S. degree in software engineering from the North Eastern University, Shenyang in 2004, M.S. degree in software engineering from Dalian University of Technology (DUT), Dalian in 2007, and Ph.D. degree in computer software and theory from DUT in 2010. Since 2014, she has been an associated professor in the School of Software Technology at the Dalian University of Technology. Her current research interests include information hiding and image processing.

Directly Revocable and Verifiable Key-Policy Attribute-based Encryption for Large Universe

Hua Ma, Ting Peng, and Zhenhua Liu
(Corresponding author: Hua Ma)

School of Mathematics and Statistics, Xidian University
Xi'an 710071, P.R. China
(Email: hma@mail.xidian.edu.cn)

(Received Dec. 22, 2015; revised and accepted Apr. 9 & May 31, 2016)

Abstract

For practical data sharing applications, many attribute-based encryption (ABE) schemes were proposed with different kinds of properties, such as supporting large universe, revocation, verification and so on. However, existing schemes seldom support these three important properties simultaneously. In this paper, we present a directly revocable and verifiable key-policy ABE scheme for large universe (DRV-KP-ABE). The new scheme supports large universe, and attributes do not need to be enumerated at stage of setup. Meanwhile, our scheme allows the trusted authority to revoke users by only updating the revocation list without interaction with non-revoked users. We use the subset difference method for revocation which greatly improves the broadcast efficiency compared with the complete subtree scheme. In addition, the proposed scheme enables the third party to update ciphertexts with public information, and the auditor assures the third party updated ciphertexts correctly. The DRV-KP-ABE scheme is selectively secure under q-type assumption in the standard model.

Keywords: Attribute-based encryption, large universe, subset difference, user revocation, verification

1 Introduction

Recently, cloud computing has attracted wide attention from all walks of life. For one thing, cloud computing can provide powerful computing capabilities for resource-constrained devices. For another thing, cloud computing allows data users to store or deliver their sensitive data in third-party servers either for ease of sharing or for cost saving. However, there have some great challenges for preserving the privacy of stored data and enforcing access control on accessing these data [14, 28, 30, 32]. Attribute-based encryption (ABE) [26], introduced by Sahai and Waters, can be viewed as the right tool solving these challenges. In ABE, a party can encrypt a document to all users who have a certain set of attributes. For example, one can encrypt a document to all hiring committee members in the computer science department. In this case the document would be encrypted to the attribute subset {"Faculty", "CS Dept.", "Hiring Committee"}, and only users with all of these three attributes can hold the corresponding private keys and thus decrypt the document.

In practical data sharing system, if users leave or be removed from the system, their access right must be deprived. However, pure ABE scheme cannot revoke these users. In order to achieve revocation, revocable ABE (R-ABE) [1] was introduced. According to how to integrate the revocation information, existing R-ABE can be divided into two categories: directly R-ABE [29] and indirectly R-ABE [4]. Direct revocation enforces revocation by the sender who specifies the revocation list while encrypting data, and it is unnecessary for users to communicate with attribute authority. On the other hand, indirect revocation enforces revocation by the key authority who sends a key update information periodically to non-revoked users such that they can update own keys. The indirect method has an advantage that senders do not need to know the revocation list. However, this approach also has a disadvantage that the key update phase could be a bottleneck since the indirect revocation requires frequently communication between the key authority and all non-revoked users. In order to eliminate this bottleneck, we consider direct user revocation in this paper.

Till now, many R-ABE schemes [1, 2, 4, 15, 23, 27, 29] were proposed. Among these schemes, most of schemes essentially employ the complete subtree (CS) scheme [15] for revocation purpose. Replacing the CS method by the subset difference (SD) technique [9, 16] can reduce the size of the ciphertext component meant for performing revocation from $O(rlog\frac{n}{r})$ to $O(r)$, where n and r denote the number of users and the number of revoked users, respectively. This can provide significant improvement in the broadcast efficiency particularly when the number of users present in the system is very large compared with the number of revoked users [9].

In fact, R-ABE alone cannot prevent revoked users from decrypting ciphertexts that were generated before revocation, since the old private key of revoked users is enough to decrypt those ciphertexts. Thus a complete solution has to support not only the revocation functionality but also the ciphertext update functionality. Sahai and Waters [25] considered the problem of updating ciphertexts in the setting of R-ABE, where the third party (*e.g.*, storage server) can update stored ciphertexts with public available information. However, in a practical application setting [13, 27], if the correctness of updated ciphertexts cannot be guaranteed, the third party may return a wrong information for some interest. This motivates us to study ABE with verifiable ciphertext delegation.

In addition, ABE can be classified into two categories depending on the size of attributes: small universe construction [3] and large universe construction [24]. In "small universe" construction, attributes are pre-specified at the stage of setup, and additional attributes cannot be added. While in "large universe" construction, the size of the attributes can be exponentially large, and a large number of new attributes can be added to the system at any time. Therefore, it seems that the scheme supporting large universe is more suitable for actual demand.

1.1 Our Contribution

We present a KP-ABE scheme that simultaneously supports user revocation, ciphertext update verification and "truly" large attribute universe. For that purpose, we make use of the technique given by Rouselakis and Waters [24] to achieve large universe construction and the one given by Shi et al. [27] in order to achieve the revocation and verification. Specially, we manage revocation for ABE by utilizing the SD mechanism [9, 16] instead of CS scheme. More precisely, we present the KP-ABE scheme with the following properties:

- The proposed scheme has a constant number of group elements in public parameters, and imposes no bound on the size of attributes used for encryption.
- In our scheme, the private key is associated with a user identity and the access policy. The ciphertext is related to the set of attributes and a revocation list. A user can decrypt a ciphertext if and only if her/his attributes satisfy the access structure and she/he is not in the revocation list. When revoking users, the trusted authority only needs to update the revocation list, without any interaction with non-revoked users.
- We use the SD method for revocation which greatly improves the broadcast efficiency and provides a smaller covering set compared with the CS scheme.
- Our scheme can delegate the third party to update ciphertexts with public available information. Meanwhile, the scheme allows the auditor to verify whether ciphertexts are updated correctly by the third party.

1.2 Organization

The rest of the paper is organized as follows. We discuss related works in Section 2 and in Section 3 we give necessary background information. We describe scheme definition and security game in Section 4. The concrete construction of DRV-KP-ABE scheme and security proof are detailed in Sections 5 and 6, respectively. The performance analysis and efficiency improvement are discussed in Section 7. Finally, we give the conclusion in Section 8.

2 Related Works

The notion of ABE, which enables fine-grained and non-interactive access control of shared data, was first introduced by Sahai and Waters [26]. Currently, there are two different and complementary forms of ABE: key-policy ABE (KP-ABE) [19] and ciphertext-policy ABE (CP-ABE) [3, 10, 21]. In KP-ABE, the ciphertext is associated with the set of attributes, and the secret key is associated with the access policy. While in CP-ABE, the idea is reversed: the user' private key is related to the set of attributes, and the access control policy on these attributes is attached to the ciphertext. Until now, many variants of ABE have been proposed to provide promising properties and functionalities, such as multi-authority ABE [18], ABE with outsourcing decryption [13, 36], traceable ABE [22], anonymous ABE [34] and so on. As a promising property, R-ABE is still an active research topic.

In 2008, Boldyreva et al. [4] constructed the first R-ABE scheme with indirect revocation, which key authority sends a key update information periodically to non-revoked users. Meanwhile, the sender doesn't care the revocation list when encrypting a message (*e.g.*, [4, 15]). Attrapadung and Imai [1] proposed a directly revocable ABE scheme, which the trusted authority revokes users by only updating the revocation list without interaction with non-revoked users. Later, they proposed a hybrid R-ABE scheme [2] which allows senders to choose the concrete revocation mode during the encryption phase. However, all of the above constructions make use of the CS method to achieve revocation. Lee et al. [16] utilized the SD scheme to achieve revocation for identity-based encryption (IBE) and pointed out that their technique for R-IBE cannot be directly applicable to construct an R-ABE scheme. Recently, Datta et al. [9] presented the first fully secure unbound R-ABE scheme in prime order bilinear groups via SD mechanism.

Note that these mechanisms in the above revocation modes only guarantee that revoked users cannot decrypt ciphertexts created after revocation. To prevent revoked users from decrypting ciphertexts that were generated before revocation, proxy re-encryption [8, 23] was introduced, which needs the interaction between the proxy and the trusted authority. Other works [25, 27, 33] considered the problem of updating ciphertexts in the setting of R-ABE, where the third party can update stored cipher-

texts without any interaction with either data owners or the trusted authority whenever the revocation event happens. In addition, Shi et al. [27] considered ciphertext update verification in the setting of directly R-ABE, where the auditor can verify whether ciphertexts were updated correctly or not.

Though these schemes have been introduced to achieve efficient revocation, they cannot support large attribute universe construction. The first large universe KP-ABE scheme in the standard model was given in [17], which is based on composite order groups. Later, several large universe ABE schemes were given in [12, 15, 24]. However, they have no consider the revocation and verification, simultaneously. Thus, it is necessary to study revocable and verifiable KP-ABE scheme for large universe.

3 Preliminaries

In this section, we give the definitions of bilinear groups, access structures and complexity assumptions. In addition, in order to achieve efficient revocation, we introduce the binary tree and SD method.

3.1 Notations

For $n \in \mathbb{N}$, we define $[n] = \{1, 2, \cdots, n\}$. Similarly, for $n_1, \cdots, n_k \in \mathbb{N} : [n_1, \cdots, n_k] = [n_1] \times [n_2] \times \cdots \times [n_k]$. When S is a set, we denote by $s \leftarrow S$ the fact that the variable s is picked uniformly at random from S. We write $s_1, s_2, \cdots, s_n \leftarrow S$ as shorthand for $s_1 \leftarrow S, s_2 \leftarrow S, \cdots, s_n \leftarrow S$. When v is a vector (of any type), we will denote by v_i the i-th element and by $\langle v, w \rangle$ the inner product of vectors v and w.

3.2 Bilinear Groups

Let $\mathbb{G}$, $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p and g be a generator of $\mathbb{G}$. A bilinear map e is a map $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ with the following properties:

1) **Bilinearity**: For all $u, v \in \mathbb{G}, a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$;

2) **Non-degeneracy**: $e(g, g) \neq 1$;

3) **Computability**: There is an efficient algorithm to compute $e(u, v)$ for all $u, v \in \mathbb{G}$.

3.3 Access Structures

In this section, we present the formal definition of access structures [24]. Let $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ be the attribute universe. An access structure on $\mathcal{P}$ is a collection $\mathbb{A}$ of non-empty sets of attributes, *i.e.*, $\mathbb{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets and the sets not in $\mathbb{A}$ are called the unauthorized sets. Additionally, an access structure is called monotone if $\forall B, C \in \mathbb{A}$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. In our construction, we only consider monotone access structures.

3.4 Linear Secret-Sharing Schemes

A linear secret-sharing scheme (LSSS) [24] can be used to represent an access control policy (M, ρ), where M is an $l \times n$ matrix which is called the share generating matrix and ρ maps a row into an attribute. A LSSS consists of two algorithms:

Share$((M, \rho), s)$**:** This algorithm is used to share secret value s to attributes. Considering a column vector $v = (s, r_2, \cdots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \cdots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $\lambda_i = M_i \cdot v$ is a share of the secret s, which belongs to the attribute $\rho(i)$.

Reconstruction$(\lambda_1, \cdots, \lambda_l, (M, \rho))$**:** This algorithm is used to reconstruct s from secret shares. Let $S \in \mathbb{A}$ be any authorized set and $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \cdots, l\}$. Then there exists coefficients $\{c_i | i \in I\}$ such that $\Sigma_{i \in I} c_i M_i = (1, 0, \cdots, 0)$, thus we have $\Sigma_{i \in I} c_i \lambda_i = s$.

3.5 Subset Difference Method

In this paper, we use the subset difference method [9, 16] to realize efficient revocation. We now give some notations (as shown in Table 1) concerning a full binary tree which is similar to those defined in literature [9]. Label the nodes in $\mathcal{BT}$ by 1 to $2n-1$ in the way that root is labeled 1, if parent is labeled i, then the left child is labeled $2i$ and the right child is labeled $2i+1$. Every member in U is assigned to a leaf node in the tree. The identifier L_i of each node in the tree is assigned as follows: Each edge in the tree is assigned with 0 or 1 depending on whether the edge connects a node to its left or right child node. The identifier L_i of a node v_i is the bit string obtained by reading all the labels of edges in the path from the root to the node v_i. The Steiner Tree $ST(R)$ is the minimal subtree of $\mathcal{BT}$ that connects all the leaf nodes in R and the root node. One example of the labeling is shown in Figure 1.

The SD scheme [9] is summarized as follows:

SD.Setup(n)**:** This algorithm takes as input the maximum number n of users (for simplicity, $n = 2^d$). It sets a full binary tree $\mathcal{BT}$ of depth d and assigns every user to a different leaf node in $\mathcal{BT}$. It outputs $\mathcal{BT}$.

SD.Assign$(\mathcal{BT}, u)$**:** This algorithm takes as input the full binary tree $\mathcal{BT}$ and a user serial number $u \in [n]$. Let $v(u)$ be the leaf node assigned to u and $path(v(u)) = (v_{k_0}, v_{k_1}, ..., v_{k_n})$ be the path from the root node v_{k_0} to the leaf node $v_{k_n} = v(u)$. For all $i, j \in \{k_0, k_1, ..., k_n\}$ such that v_j is a descendant of v_i, it adds $S_{i,j}$ defined by two nodes v_i and v_j in the path into a private set PV_u. Finally, it outputs PV_u.

SD.Cover$(\mathcal{BT}, R)$**:** This algorithm takes as input the binary tree $\mathcal{BT}$ and a revoked set R of users. Then it

Table 1: Notations

Notation	Significance
n	The maximum number of users in the system
$\mathcal{BT}$	A full binary tree with n leaf nodes
U	A serial number set of user
R	A serial number set of revoked user
v_i	A node in $\mathcal{BT}$ for any $i, i \in [2n-1]$
d_i	The depth of the node v_i
T_i	A subtree of $\mathcal{BT}$ that is rooted at v_i, where $v_i \in \mathcal{BT}$
$T_{i,j}$	A subtree $T_i - T_j$ for any $v_i, v_j \in \mathcal{BT}$ such that v_i is the ancestor of v_j
S_i	The set of leaf nodes of T_i
$S_{i,j}$	The set of leaf nodes of $T_{i,j}$
L_i	An identifier for a node v_i in $\mathcal{BT}$, a fixed and unique string
$(L_i\|\|d_j)$	The integer representation of the string formed by concatenating d_j with L_i
$(L_i\|\|L_j)$	The integer representation of the string obtained by concatenating L_j with L_i
$ST(R)$	The Steiner Tree induced by a subset R and the root node

computes CV_R iteratively by removing nodes from $ST(R)$ until $ST(R)$ only has a single node as follows (an example is given in Figure 1):

1) Find two leaves v_i and v_j in $ST(R)$ such that the least-common-ancestor (lca) v of v_i and v_j does not contain any other leaf node of $ST(R)$ in its subtree. Let v_l and v_k be the two children of v such that v_i is a descendant of v_l, and v_j is a descendant of v_k. (If there is only one leaf node, make $v_i = v_j$ to be the leaf, v to be the root of $ST(R)$ and $v_l = v_k = v$.)
2) If $v_l \neq v_i$, $CV_R = CV_R \bigcup S_{l,i}$; if $v_k \neq v_j$, $CV_R = CV_R \bigcup S_{k,j}$.
3) Remove from $ST(R)$ all the descendants of v and make it a leaf.

SD.Match(CV_R, PV_u)**:** This algorithm takes as input a covering set $CV_R = \{S_{i,j}\}$ and a private set $PV_u = \{S_{i',j'}\}$. It obtains $(S_{i,j}, S_{i',j'})$ such that $S_{i,j} \in CV_R, u \in S_{i,j}$, and $S_{i',j'} \in PV_u$, or obtains $\perp$.

For all $n = 2^d$, let $\mathcal{BT} \leftarrow$ **SD.Setup**(n), $PV_u \leftarrow$ **SD.Assign**$(\mathcal{BT}, u)$, $CV_R \leftarrow$ **SD.Cover**$(\mathcal{BT}, R)$. The correctness of the SD scheme is defined as follows:

1) If $u \notin R$, then **SD.Match**$(CV_R, PV_u) = (S_{i,j}, S_{i',j'})$;
2) If $u \in R$, then **SD.Match**$(CV_R, PV_u) = \perp$.

As shown in Figure 1, given $R = \{u_2, u_3, u_4, u_6\}$. According to the algorithm **SD.Cover**$(\mathcal{BT}, R)$, choose v_5 as a lca of v_{10} and v_{11}. Get v_5 as a leaf, v_2 is a lca of v_9 and v_5. Get v_2 as a leaf, $CV_R = \{S_{4,9}\}$, the root is a lca of v_{13} and v_2, $CV_R = \{S_{4,9}, S_{3,13}\}$. For $u_1 \notin R$, $PV_{u_1} = \{S_{1,2}, S_{1,4}, S_{1,8}, S_{2,4}, S_{2,8}, S_{4,8}\}$, run the algorithm **SD.Match**(CV_R, PV_{u_1}) and output $(S_{4,9}, S_{4,8})$. Similarly, the algorithm **SD.Match**(CV_R, PV_{u_8}) outputs $(S_{3,13}, S_{3,15})$. For $u \in R$, the algorithm **SD.Match**(CV_R, PV_u) outputs $\perp$.

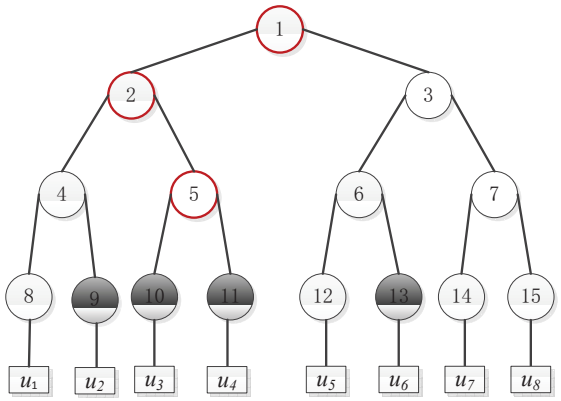


Figure 1: An Example of Binary Tree $\mathcal{BT}$

Lemma 1. *Let n be the number of leaf nodes and r be the size of a revoked set. The size of a private set is $O(log^2 n)$ and the size of a covering set is at most $2r-1$ in the SD scheme [9].*

3.6 Complexity Assumption

For our KP-ABE construction we will use a q-type assumption [24], which is similar to the Decisional Bilinear Diffie-Hellman Assumption augmented with q parameters b_i. The assumption is defined via the following game between a challenger and an attacker:

Initially, the challenger inputs the security parameter and picks a random group element $g \leftarrow \mathbb{G}$ and $q+3$ random exponents $a, b, c, b_1, b_2, \cdots, b_q \leftarrow \mathbb{Z}_p$. Then he sends the group description $(p, \mathbb{G}, \mathbb{G}_T, e)$ and all of the

following terms to the attacker:

$$g, g^a, g^b, g^c, g^{(ac)^2},$$
$$g^{b_i}, g^{acb_i}, g^{ac/b_i}, g^{a^2cb_i}, g^{b/b_i^2}, g^{b^2/b_i^2}, \forall i \in [q],$$
$$g^{acb_i/b_j}, g^{bb_i/b_j^2}, g^{abcb_i/b_j}, g^{(ac)^2b_i/b_j}, \forall i, j \in [q], i \neq j.$$

The challenger flips a random coin $\mu \leftarrow \{0,1\}$ and if $\mu = 0$ he gives the term $e(g,g)^{abc}$ to the attacker. Otherwise he gives a random term $\mathcal{R} \leftarrow \mathbb{G}_T$. Finally the attacker outputs a guess $\mu' \in \{0,1\}$.

Definition 1. *We say that the q-type assumption holds if all PPT attackers have at most a negligible advantage in λ in the above security game, where the advantage is defined as*

$$Adv = |\Pr[\mu' = \mu] - 1/2|.$$

4 Syntax and Security

4.1 Algorithms

A directly revocable and verifiable key-policy attribute-based encryption (KP-ABE) scheme for large universe consists of the following six algorithms:

Setup$(1^\lambda, n) \rightarrow (pp, msk)$**:** Input a security parameter λ and the maximum number n of users. The algorithm obtains $\mathcal{BT}$ by running **SD.Setup**(n) and outputs the public parameters pp and the master secret key msk. We assume that the public parameters contain a description of the attribute universe $\mathcal{N}$.

Extract$(msk, ID, \mathbb{A}) \rightarrow sk$**:** Given a user identity ID, choose an unassigned leaf node $v(u)$ in $\mathcal{BT}$ at random and assign ID to the leaf node $v(u)$, where $u \in [n]$ is a serial number that is assigned to ID. (For convenience, a serial number and the corresponding user identity can be exchanged in this paper.) The algorithm runs **SD.Assign**$(\mathcal{BT}, u)$ to obtain $PV_u = \{S_{i,j}\}$. Input the master secret key msk and an access structure $\mathbb{A}$ on $\mathcal{N}$. The algorithm generates a secret key corresponding to $\mathbb{A}$ and ID.

Encrypt$(pp, m, S, R) \rightarrow ct$**:** Input the public parameters pp, a plaintext message m, and a set of attributes $S \subseteq \mathcal{N}$, as well as the current revocation list R. The algorithm obtains the cover set CV_R by executing **SD.Cover**$(\mathcal{BT}, R)$ and outputs the ciphertext ct.

Decrypt$(sk, ct) \rightarrow m$**:** Input a secret key sk, and a ciphertext ct, if the attribute set S satisfies the ciphertext policy and user identity $ID \notin R$, then user obtains $(S_{i,j}, S_{\tilde{i},\tilde{j}})$ by running **SD.Match**(CV_R, PV_u) such that $S_{i,j} \in CV_R, S_{\tilde{i},\tilde{j}} \in PV_u$ and $(i = \tilde{i}) \wedge (d_j = d_{\tilde{j}}) \wedge (j \neq \tilde{j})$ and recovers a message m. Otherwise, return $\perp$.

CTUpdate$(ct, R', pp) \rightarrow ct'$**:** Input ct, the latest revocation list R' and pp, the third party updates the original ciphertext ct to ct' associated with R'.

Verify$(ct, ct') \rightarrow \{0,1\}$**:** After the third party finished the ciphertexts updating, the auditors will verify whether ciphertexts were updated correctly from some prior revocation lists to the current revocation lists or not. Then he outputs 1 if the update is correct, and 0 otherwise.

Let $(pp, msk) \leftarrow$ **Setup**(1^λ), $ct \leftarrow$ **Encrypt**(pp, m, S, R), $ct' \leftarrow$ **CTUpdate**(ct, R', pp), $sk \leftarrow$ **Extract**$(msk, ID, \mathbb{A})$. For correctness, we require the following conditions always hold:

1) $1 \leftarrow$ **Verify**(ct, ct').
2) If $ID \notin R$ and the attribute set S satisfies the ciphertext policy $\mathbb{A}$, the algorithm returns $m \leftarrow$ **Decrypt**(sk, ct). Otherwise, return $\perp$.
3) If $ID \notin R'$ and the attribute set S satisfies the ciphertext policy $\mathbb{A}$, the algorithm returns $m \leftarrow$ **Decrypt**(sk, ct'). Otherwise, return $\perp$.

4.2 Selective Security Game

Similar to the security model in the literature [27], since the updated ciphertexts have the same distribution as original ciphertexts, we only consider the security of original ciphertexts. We now describe the security model of our system by the following game between a challenger and an attacker:

Initialization: In this phase, the attacker $\mathcal{A}$ declares the challenge attribute set S^* and a revocation list R^*, which he will try to attack, and sends them to the challenger $\mathcal{B}$.

Setup: The challenger runs **Setup**(1^λ) to obtain the public parameters pp and sends pp to the attacker $\mathcal{A}$.

Query phase 1: In this phase, the attacker $\mathcal{A}$ can adaptively issue extract queries for secret keys related to several tuples $(ID, \mathbb{A})$.

- If $S \in \mathbb{A}$ and $ID \notin R^*$, then $\mathcal{B}$ will abort.
- Otherwise, the challenger generates a secret key related to $(ID, \mathbb{A})$ for the attacker $\mathcal{A}$.

Challenge: The attacker $\mathcal{A}$ declares two equal-length plaintexts m_0 and m_1 and submits them to the challenger. The challenger flips a random coin $\nu \in \{0,1\}$ and calls **Encrypt**$(m_\nu, S^*, R^*) \rightarrow ct$. He sends ct to the attacker.

Query phase 2: Phase 2 is the same as Phase 1.

Guess: The attacker $\mathcal{A}$ outputs his guess $\nu' \in \{0,1\}$ for ν.

Definition 2. *A KP-ABE scheme is selectively secure if all PPT attackers have at most a negligible advantage in λ in the above security game, where the advantage of an attacker is defined as*

$$Adv = |\Pr[\nu = \nu'] - 1/2|.$$

4.3 Verifiability Game

In this subsection, we give the security model of verifiability based on the literature [27, 36].

Initialization: In this phase, the attacker $\mathcal{A}$ chooses an attribute set S^* and sends it to the challenger.

Setup: The challenger runs $\textbf{Setup}(1^\lambda)$ to obtain the public parameters pp and sends pp to the attacker $\mathcal{A}$.

Query phase 1: In this phase, the attacker $\mathcal{A}$ can adaptively issue queries:

1) **Extract query**: Input several tuples $(ID, \mathbb{A})$, the challenger generates a secret key sk related to $(ID, \mathbb{A})$ for the attacker $\mathcal{A}$.
2) **Verification query**: Input updated ciphertexts ct^*, the challenger returns γ to $\mathcal{A}$ by running $\textbf{Verify}(ct, ct^*) \to \gamma$.

Challenge: The attacker $\mathcal{A}$ selects a plaintext m and two revocation lists R and R^*, where $R \subset R^*$, and submits them to the challenger. The challenger calls $\textbf{Encrypt}(m, S^*, R) \to ct$. He sends ct to the attacker.

Guess: The attacker $\mathcal{A}$ outputs a updated ciphertext ct^* to the challenger associated with the revocation list R^*. The attacker $\mathcal{A}$ wins this game if $\textbf{Verify}(ct, ct^*) \to 1$ and the distribution of ct^* and ct' are computationally distinguishable, where $\textbf{Update}(ct, R^*, pp) \to ct'$ produced by the challenger.

Definition 3. *We say that the proposed scheme achieves update verifiability if the advantage that any $\mathcal{A}$ wins the verifiability game is negligible in security parameter λ.*

5 Our Construction

We construct a directly revocable and verifiable KP-ABE scheme for large universe based on the techniques in paper [24, 27]. Different from the scheme in the literature [27], we use the SD scheme to manage revocation for ABE. The SD method provides a smaller covering set compared with the CS scheme, particularly when the number of users present in the system is very large. In addition, we also use the method given by Lee et al. [16] to solve the complex key assignment problem of the SD mechanism. Now we describe the scheme as follows.

$\textbf{Setup}(1^\lambda, n) \to (pp, msk)$: Input a security parameter λ and the maximum number n of users. The algorithm obtains $\mathcal{BT}$ by running $\textbf{SD.Setup}(n)$. Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. The attribute universe is $\mathcal{N} = \mathbb{Z}_p$. Select the random terms $g, z, h, w, f \leftarrow \mathbb{G}$ and $\alpha, \beta \leftarrow \mathbb{Z}_p$. Let $v = g^\beta$. The public parameters are published as

$$pp = (g, z, h, w, v, f, e(g, g)^\alpha).$$

The authority sets $msk = (\alpha, \beta)$ as the master secret key.

$\textbf{Extract}(msk, ID, (M, \rho)) \to sk$: The algorithm picks $y = (\alpha_1, y_2, \cdots, y_n)^\top$ and sets α_2 such that $\alpha = \alpha_1 + \alpha_2 (\text{mod } p)$, where $\alpha_1, y_2, \cdots, y_n \leftarrow \mathbb{Z}_p$ and α_1 is the secret to be shared among the shares. The vector of the shares is

$$M \cdot y = (\lambda_1, \lambda_2, \cdots, \lambda_l)^\top.$$

Select l random exponents $k_1, k_2, \cdots, k_l \leftarrow \mathbb{Z}_p$ and for every $\tau \in [l]$

$$K_{\tau,0} = g^{\lambda_\tau} w^{k_\tau}, K_{\tau,1} = (z^{\rho(\tau)} h)^{-k_\tau}, K_{\tau,2} = g^{k_\tau}.$$

Given user identity ID, choose an unassigned leaf node $v(u)$ in $\mathcal{BT}$ at random and assign ID to the leaf node $v(u)$, where $u \in [n]$ is a serial number that is assigned to ID. Next the algorithm obtains PV_u by running $\textbf{SD.Assign}(\mathcal{BT}, u)$. For each $S_{i,j} \in PV_u$, pick $\theta_{i,j,1}, \theta_{i,j,2} \leftarrow \mathbb{Z}_p$ and compute

$$D_{i,j,1} = g^{\alpha_2} v^{(L_i || d_j)\theta_{i,j,1}} f^{\theta_{i,j,2}}, E_{i,j,1} = g^{-\theta_{i,j,1}},$$
$$D_{i,j,2} = (f^{(L_i || L_j)} v)^{\theta_{i,j,2}}, E_{i,j,2} = g^{-\theta_{i,j,2}}.$$

The secret key is

$$sk = (PV_u, ID, (M, \rho), \{K_{\tau,0}, K_{\tau,1}, K_{\tau,2}\}_{\tau \in [l]}, \{D_{i,j,t}, E_{i,j,t}\}_{S_{i,j} \in PV_u, t=1,2}).$$

$\textbf{Encrypt}(m, S = \{A_1, A_2, \cdots, A_k\} \subseteq Z_p, R, pp) \to ct$: The algorithm picks $k+1$ random exponents $s, r_1, r_2, \cdots, r_k \leftarrow \mathbb{Z}_p$ and computes

$$C = m \cdot e(g, g)^{\alpha s}, C_0 = g^s,$$
$$\forall \tau \in [k], C_{\tau,1} = g^{r_\tau}, C_{\tau,2} = (z^{A_\tau} h)^{r_\tau} w^{-s}.$$

Given the revoked user serial number set $R \subseteq [n]$, the algorithm obtains the cover set CV_R by executing $\textbf{SD.Cover}(\mathcal{BT}, R)$. For each $S_{i,j} \in CV_R$, set

$$C_{i,j,1} = v^{(L_i || d_j) s}, C_{i,j,2} = (f^{(L_i || L_j)} v)^s.$$

The ciphertext is

$$ct = (S, R, CV_R, C, C_0, \{C_{\tau,1}, C_{\tau,2}\}_{\tau \in [k]}, \{C_{i,j,t}\}_{S_{i,j} \in CV_R, t=1,2}).$$

$\textbf{Decrypt}(sk, ct) \to m$: Given sk and ct, the decryption can be done as follows:

- If user identity $ID \in R$ or the attribute set S does not satisfy the ciphertext policy, then return $\perp$.
- Otherwise, proceed as follows. Suppose the set S satisfies the access structure (M, ρ) and $ID \notin R$. Let $I = \{i : \rho(i) \in S\}$. There exists constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} \omega_i M_i =$

$(1, 0, \cdots, 0)$. Since $ID \notin R$, then user obtains $(S_{i,j}, S_{\tilde{i},\tilde{j}})$ by running **SD.Match**(CV_R, PV_u) such that $S_{i,j} \in CV_R, S_{\tilde{i},\tilde{j}} \in PV_u$ and $(i = \tilde{i}) \wedge (d_j = d_{\tilde{j}}) \wedge (j \neq \tilde{j})$.
Then the user calculates

$$\begin{aligned} B =& \prod_{i \in I} (e(C_0, K_{i,0}) e(C_{\tau,1}, K_{i,1}) \\ & \cdot e(C_{\tau,2}, K_{i,2}))^{\omega_i}, \\ B' =& e(C_0, D_{\tilde{i},\tilde{j},1}) e(C_{i,j,1}, E_{\tilde{i},\tilde{j},1}) [e(C_0, D_{\tilde{i},\tilde{j},2}) \\ & \cdot e(C_{i,j,2}, E_{\tilde{i},\tilde{j},2})]^{\frac{-1}{(L_{\tilde{i}}||L_{\tilde{j}}) - (L_i||L_j)}}, \\ m =& C/(BB'), \end{aligned}$$

where τ is the index of the attribute $\rho(i)$ in S (it depends on i).

CTUpdate$(ct, R', pp) \rightarrow ct'$**:** Given the latest revoked user serial number set R', the cover set $CV_{R'}$ is obtained by running **SD.Cover**$(\mathcal{BT}, R')$. For $S_{i',j'} \in CV_{R'}$, there are two cases.

Case 1: If there exists $S_{i,j} \in CV_R$ such that $(i = i') \wedge (j = j')$ then set $ct' = ct$.

Case 2: Otherwise, there exists $S_{i,j} \in CV_R$ such that v_i is an ancestor of $v_{i'}$ or $v_i = v_{i'}$. Choose the random exponent $s' \leftarrow \mathbb{Z}_p$ and set

$$\begin{aligned} C_{i',j',1} =& (C_{i,j,1})^{\frac{(L_{i'}||d_{j'})}{(L_i||d_j)}} \cdot v^{(L_{i'}||d_{j'})s'}, \\ C_{i',j',2} =& (C_{i,j,2})^{\frac{(L_{i'}||L_{j'})}{(L_i||L_j)}} \cdot f^{(L_{i'}||L_{j'})s'} \\ & (C_{i,j,1})^{\frac{1}{(L_i||d_j)}(1 - \frac{(L_{i'}||L_{j'})}{(L_i||L_j)})} \cdot v^{s'}. \end{aligned}$$

Finally, let $C' = C \cdot e(g,g)^{\alpha s'}, C'_0 = C_0 \cdot g^{s'}, C'_{\tau,1} = C_{\tau,1}, C'_{\tau,2} = C_{\tau,2} \cdot w^{-s'}, \forall \tau \in [k]$. The updated ciphertext is

$$\begin{aligned} ct' =& (S, R', CV_{R'}, C', C'_0, \{C'_{\tau,1}, C'_{\tau,2}\}_{\tau \in [k]}, \\ & \{C'_{i',j',t}\}_{S_{i',j'} \in CV_{R'}, t=1,2}). \end{aligned}$$

Verify$(ct, ct') \rightarrow \{0, 1\}$**:** The verification can be done as follows:

- Verify whether the following equation holds:

$$\begin{aligned} & e(g, C'_{\tau,2} C'/C) e(C'_0, w) \\ & = e(C'_{\tau,1}, z^{A_\tau} h) e(C'_0/C_0, e(g,g)^{\alpha}), \end{aligned}$$

where $\tau \in [k]$.
If not, then output 0. Otherwise, proceed to the following step.

- Figure out $S_{i'_1,j'_1}, \cdots, S_{i'_\eta,j'_\eta}$ such that $S_{i'_k,j'_k} \in CV_{R'} - CV_R$, where $k \in \{1, \cdots, \eta\}$, select $c_1, \cdots, c_\eta \leftarrow \mathbb{Z}_p$ and verify

$$\begin{aligned} & e(C'_0, \prod_{k=1}^{\eta} (v^{(L_{i'_k}||d_{j'_k})})^{c_k}) = e(g, \prod_{k=1}^{\eta} (C'_{i'_k,j'_k,1})^{c_k}), \\ & e(C'_0, \prod_{k=1}^{\eta} (f^{(L_{i'_k}||L_{j'_k})} v)^{c_k}) = e(g, \prod_{k=1}^{\eta} (C'_{i'_k,j'_k,2})^{c_k}). \end{aligned}$$

If the above equations do not hold, then return 0. Otherwise, return 1.

6 Security Proof

Since the updated ciphertexts has the same distribution as the original ciphertexts. We only prove the security related to the original ciphertexts and updated verifiability.

Theorem 1. *If the q-type assumption holds, then all PPT attackers with a challenge attribute set of size k, where $k \leq q$, have a negligible advantage in selectively breaking our scheme.*

Proof. To prove the theorem we will assume that there exists a PPT attacker $\mathcal{A}$ with a challenge attribute set, which has a non-negligible advantage $Adv_{\mathcal{A}}$ in selectively breaking our scheme. Using this attacker we will build a PPT challenger $\mathcal{B}$ that attacks the q-type assumption with a non-negligible advantage.

Initialization: Initially, $\mathcal{B}$ receives the given terms from the assumption, an attribute set $S^* = \{A_1^*, A_2^*, \cdots, A_k^*\} \subseteq N$ and a revocation list R^*.

Setup: Now, the challenger $\mathcal{B}$ provides the public parameters of the system for $\mathcal{A}$. In order to do that $\mathcal{B}$ implicitly sets the master secret key of the scheme to be $\alpha = ab$ and $\beta = b + d$, where a, b are set in the assumption and $d \leftarrow \mathbb{Z}_p$. $\mathcal{B}$ picks the random terms $f, \tilde{z}, \tilde{h} \leftarrow \mathbb{Z}_p$ and gives the following terms to $\mathcal{A}$:

$$\begin{aligned} g &= g, \\ z &= g^{\tilde{z}} \cdot \prod_{i \in [k]} g^{b/b_i^2}, \\ h &= g^{\tilde{h}} \cdot \prod_{i \in [k]} g^{ac/b_i} \cdot \prod_{i \in [k]} (g^{b/b_i^2})^{-A_i^*}, \\ w &= g^a, \\ v &= g^{\beta} = g^{b+d}, \\ e(g,g)^{\alpha} &= e(g^a, g^b). \end{aligned}$$

Phase 1: In this phase, the attacker $\mathcal{A}$ issues extract queries for secret keys related to several tuples $(ID, (M, \rho))$, and the challenger proceeds as follows:

Case 1: If $S^* \in (M, \rho)$ and $ID \notin R^*$, then $\mathcal{B}$ will abort.

Case 2: If $ID \in R^*$, the challenger selects $\alpha_1 \in \mathbb{Z}_p$ and computes $K_{\tau,0}, K_{\tau,1}, K_{\tau,2}$ based on the algorithm **Extract**$(msk, (M, \rho))$. Next the challenger chooses an unassigned leaf node $v(u)$ in $\mathcal{BT}$ at random and assigns ID to the leaf node $v(u)$, where $u \in [n]$ is a serial number that is assigned to ID. Then he obtains PV_u by running **SD.Assign**$(\mathcal{BT}, u)$. For each $S_{i,j} \in PV_u$,

choose $\theta'_{i,j,1}, \theta_{i,j,2} \in \mathbb{Z}_p$ at random and set

$$\begin{aligned}
D_{i,j,1} =& g^{-\alpha_1} g^{b(L_i||d_j)\theta'_{i,j,1}} f^{\theta_{i,j,2}} \\
& \cdot g^{d(L_i||d_j)(\theta'_{i,j,1} - \frac{a}{(L_i||d_j)})}, \\
E_{i,j,1} =& g^{-\theta_{i,j,1}} = g^{\frac{a}{(L_i||d_j)} - \theta'_{i,j,1}}, \\
D_{i,j,2} =& (f^{(L_i||L_j)} v)^{\theta_{i,j,2}}, \\
E_{i,j,2} =& g^{-\theta_{i,j,2}}
\end{aligned}$$

by implicitly defining

$$\begin{aligned}
\alpha_2 &= \alpha - \alpha_1 = ab - \alpha_1, \\
\theta_{i,j,1} &= \theta'_{i,j,1} - \frac{a}{(L_i||d_j)}.
\end{aligned}$$

Case 3: If $S^* \notin (M, \rho)$, the challenger selects $\alpha_2 \in \mathbb{Z}_p$ and computes $D_{i,j,1}, E_{i,j,1}, D_{i,j,2}, E_{i,j,2}$ based on the algorithm **Extract**(msk, ID). Since $S^* \notin (M, \rho)$, there exists a vector $\omega = (\omega_1, \omega_2, \cdots, \omega_n)^\top \in \mathbb{Z}_p^n$ such that $\omega_1 = 1$ and $\langle M_\tau, \omega \rangle = 0$ for all $\tau \in [l]$ such that $\rho(\tau) \in S^*$. The vector y that will be shared is implicitly

$$y = (\alpha - \alpha_2)\omega + (0, \tilde{y}_2, \tilde{y}_3, \cdots, \tilde{y}_n)^\top,$$

where $\tilde{y}_2, \tilde{y}_3, \cdots, \tilde{y}_n \leftarrow \mathbb{Z}_p$. For each row $\tau \in [l]$ the share is

$$\lambda_\tau = \langle M_\tau, y \rangle = (ab - \alpha_2)\langle M_\tau, \omega \rangle + \tilde{\lambda}_\tau.$$

Then the challenger computes $K_{\tau,0}, K_{\tau,1}, K_{\tau,2}$ as follows:

- If $\rho(\tau) \in S^*$: In this case $\lambda_\tau = \langle M_\tau, y \rangle = \tilde{\lambda}_\tau = \langle M_\tau, (0, \tilde{y}_2, \tilde{y}_3, \cdots, \tilde{y}_n)^\top \rangle$; hence its value is known to the challenger. Pick $K_\tau \in \mathbb{Z}_p$ and output the terms $K_{\tau,0}, K_{\tau,1}, K_{\tau,2}$ as in the algorithm **Extract**.
- Otherwise $\rho(\tau) \notin S^*$: Pick $\tilde{k}_\tau \in \mathbb{Z}_p$ and compute

$$\begin{aligned}
K_{\tau,0} =& g^{\lambda_\tau} w^{k_\tau} \\
=& g^{\tilde{\lambda}_\tau - \alpha_2 \langle M_\tau, \omega \rangle} \\
& \cdot \prod_{i \in [k]} (g^{a^2 c b_i})^{\frac{\langle M_\tau, \omega \rangle}{(\rho(\tau) - A_i^*)}} \cdot w^{\tilde{k}_\tau}, \\
K_{\tau,1} =& (z^{\rho(\tau)} h)^{-k_\tau} \\
=& (g^b)^{\langle M_\tau, \omega \rangle (\rho(\tau)\tilde{z} + \tilde{h})} \cdot (z^{\rho(\tau)} h)^{-\tilde{k}_\tau} \\
& \cdot \prod_{i \in [k]} (g^{acb_i})^{\frac{-(\rho(\tau)\tilde{z} + \tilde{h})\langle M_\tau, \omega \rangle}{(\rho(\tau) - A_i^*)}} \\
& \cdot \prod_{(i,j) \in [k,k]} (g^{(ac)^2 b_j / b_i})^{\frac{-\langle M_\tau, \omega \rangle}{(\rho(\tau) - A_j^*)}} \\
& \cdot \prod_{i \in [k]} (g^{b^2 / b_i^2})^{\langle M_\tau, \omega \rangle (\rho(\tau) - A_i^*)} \\
& \cdot \prod_{(i,j) \in [k,k]} (g^{\frac{abcb_j}{b_i^2}})^{\frac{-\langle M_\tau, \omega \rangle (\rho(\tau) - A_i^*)}{(\rho(\tau) - A_j^*)}} \\
& \cdot \prod_{i \in [k]} (g^{\frac{abc\langle M_\tau, \omega \rangle}{b_i}}), \\
K_{\tau,2} =& g^{k_\tau} \\
=& (g^b)^{-\langle M_\tau, \omega \rangle} \cdot g^{\tilde{k}_\tau} \\
& \cdot \prod_{i \in [k]} (g^{acb_i})^{\frac{\langle M_\tau, \omega \rangle}{(\rho(\tau) - A_i^*)}}
\end{aligned}$$

by implicitly setting

$$k_\tau = -b\langle M_\tau, \omega \rangle + \sum_{i \in [k]} \frac{acb_i \langle M_\tau, \omega \rangle}{\rho(\tau) - A_i^*} + \tilde{k}_\tau.$$

Challenge: The attacker will output a pair of messages (m_0, m_1) of the same length. The challenger flips a random coin $\nu \leftarrow \{0, 1\}$ and implicitly sets $s = c$ from the q-type assumption. Also, set $r_\tau = b_\tau$ for every level $\tau \in [k]$. These parameters are properly distributed since $c, b_1, b_2, \cdots, b_q$ are information-theoretically hidden from the attacker's view. Now the challenger computes the following terms using the assumption:

$$\begin{aligned}
C =& m_\nu T,\ C_0 = g^s = g^c,\ C_{\tau,1} = g^{r_\tau} = g^{b_\tau}, \\
C_{\tau,2} =& (z^{A_\tau^*} h)^{r_\tau} w^{-s} \\
=& g^{b_\tau(\tilde{z} A_\tau^* + \tilde{h})} g^{-ac} \\
& \cdot \prod_{i \in [k]} g^{acb_\tau / b_i} \prod_{i \in [k]} g^{bb_\tau (A_\tau^* - A_i^*)/b_i^2} \\
=& (g^{b_\tau})^{\tilde{z} A_\tau^* + \tilde{h}} \prod_{i \in [k], i \neq \tau} g^{acb_\tau / b_i} \\
& \cdot \prod_{i \in [k], i \neq \tau} (g^{bb_\tau / b_i^2})^{A_\tau^* - A_i^*}.
\end{aligned}$$

For each $S_{i,j} \in CV_{R^*}$, set

$$\begin{aligned}
C_{i,j,1} &= v^{(L_i||d_j)s} = v^{(L_i||d_j)c}, \\
C_{i,j,2} &= (f^{(L_i||L_j)} v)^s = (f^{(L_i||L_j)} v)^c.
\end{aligned}$$

The challenger sends the ciphertext ct to $\mathcal{A}$, where

$$\begin{aligned}
ct =& (S^*, R^*, CV_{R^*}, C, C_0, \{C_{\tau,1}, C_{\tau,2}\}_{\tau \in [k]}, \\
& \{C_{i,j,t}\}_{S_{i,j} \in CV_{R^*}, t=1,2}).
\end{aligned}$$

Phase 2: Same as Phase 1.

Guess: $\mathcal{A}$ outputs a guess ν' of ν. If $\nu' = \nu$ the challenger outputs $\mu' = 0$ and guesses the challenge term is $T = e(g, g)^{abc}$. Otherwise, the challenger outputs $\mu' = 1$ and guesses T is a random group element $\mathcal{R}$. It is obvious that the generation of public parameters and private keys is identical to the actual scheme.

In the case where $\mu = 1$ the attacker gains no information about ν. Therefore, we have $\Pr[\nu \neq \nu' | \mu = 1] = \frac{1}{2}$. Since the challenger guesses $\mu' = 1$ when $\nu \neq \nu'$, we have $\Pr[\mu = \mu' | \mu = 1] = \frac{1}{2}$.

If $\mu = 0$ the attacker can see an encryption of M_ν. In this situation, the attacker's advantage is ε by definition. Thus we have $\Pr[\nu = \nu' | \mu = 0] = \frac{1}{2} + \varepsilon$. Since the challenger guesses $\mu' = 0$ when $\nu = \nu'$, we have $\Pr[\mu = \mu' | \mu = 0] = \frac{1}{2} + \varepsilon$.

Finally, the overall advantage of the challenger in the q-type game is

$$\frac{1}{2}\Pr[\mu = \mu' | \mu = 0] - \frac{1}{2} + \frac{1}{2}\Pr[\mu = \mu' | \mu = 1] = \frac{1}{2}\varepsilon.$$

□

Theorem 2. *The proposed scheme is verifiable, if no polynomial time attacker $\mathcal{A}$ can get non-negligible advantage in the verifiability game defined in Section 4.3.*

Proof. We show that any polynomial time attacker $\mathcal{A}$ presents an incorrect updated ciphertext and succeeds in the verification with negligible probability.

The challenger proceeds the verifiability game, where $\mathcal{A}$ provides the attribute set S^*. The challenger runs $\mathbf{Setup}(1^\lambda)$ to obtain the public parameters pp and sends pp to the attacker $\mathcal{A}$. In the challenge phase, the challenger obtains a plaintext m and two revocation lists R and R^* from $\mathcal{A}$, where $R \subset R^*$. The challenger returns ct to the attacker by running $\mathbf{Encrypt}(m, S^*, R) \rightarrow ct$. $\mathcal{A}$ outputs the updated ciphertext ct^* to the challenger associated with the revocation list R^*.

Suppose that ct^* succeeds in the verification. That is, $\mathbf{Verify}(ct, ct^*) \rightarrow 1$. Let us consider the probability of $\mathcal{A}$ cheating with incorrect updated ciphertext.

Suppose the original ciphertext

$$\begin{aligned} ct =& (S^*, R, CV_R, C, C_0, \{C_{\tau,1}, C_{\tau,2}\}_{\tau\in[k]}, \\ & \{C_{i,j,t}\}_{S_{i,j}\in CV_R, t=1,2}). \end{aligned}$$

The updated ciphertext

$$\begin{aligned} ct^* =& (S^*, R^*, CV_{R^*}, C^*, C_0^*, \{C_{\tau,1}^*, C_{\tau,2}^*\}_{\tau\in[k]}, \\ & \{C_{i',j',t}^*\}_{S_{i',j'}\in CV_{R^*}, t=1,2}), \\ ct' =& (S^*, R^*, CV_{R^*}, C', C_0', \{C_{\tau,1}', C_{\tau,2}'\}_{\tau\in[k]}, \\ & \{C_{i',j',t}'\}_{S_{i',j'}\in CV_{R^*}, t=1,2}), \end{aligned}$$

where ct^* is the updated ciphertext returned by the attacker $\mathcal{A}$, when the revocation list R is changed to R^*, such that $R \subset R^*$, ct' is the updated ciphertext outputted by algorithm **update**.

Since $\mathbf{Verify}(ct, ct^*) \rightarrow 1$, then the following equations hold:

$$e(g, \frac{C_{\tau,2}^* C^*}{C}) e(C_0^*, w) = e(C_{\tau,1}^*, z^{A_\tau} h) e(\frac{C_0^*}{C_0}, e(g,g)^\alpha), \quad (1)$$

and

$$\begin{aligned} e(C_0^*, \prod_{k=1}^{\eta} (v^{(L_{i'_k} || d_{j'_k})})^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}^*)^{c_k}), \quad (2) \\ e(C_0^*, \prod_{k=1}^{\eta} (f^{(L_{i'_k} || L_{j'_k})} v)^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,2}^*)^{c_k}), \end{aligned}$$

where $\tau \in [k], S_{i'_k,j'_k} \in CV_{R^*} - CV_R, 1 \leq k \leq \eta$ and $c_1, \cdots, c_\eta \leftarrow \mathbb{Z}_p$ are randomly selected by the challenger and unknown to $\mathcal{A}$.

Since $\mathbf{Verify}(ct, ct') \rightarrow 1$ (due to the correctness of the scheme), the following equations should hold:

$$e(g, \frac{C_{\tau,2}' C'}{C}) e(C_0', w) = e(C_{\tau,1}', z^{A_\tau} h) e(\frac{C_0'}{C_0}, e(g,g)^\alpha), \quad (3)$$

and

$$\begin{aligned} e(C_0', \prod_{k=1}^{\eta} (v^{(L_{i'_k} || d_{j'_k})})^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}')^{c_k}), \quad (4) \\ e(C_0', \prod_{k=1}^{\eta} (f^{(L_{i'_k} || L_{j'_k})} v)^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,2}')^{c_k}), \end{aligned}$$

where $\tau \in [k], \{S_{i'_k,j'_k}\}_{1\leq k\leq\eta}$ and $\{c_i\}_{1\leq i\leq\eta}$ are the same as the above.

According to Equations (1) and (3), we have $C' = C^*, C_0' = C_0^*, C_{\tau,1}' = C_{\tau,1}^*, C_{\tau,2}' = C_{\tau,2}^*$. In order to prove $ct' = ct^*$, we need to prove $\forall k, 1 \leq k \leq \eta, C_{i'_k,j'_k,1}' = C_{i'_k,j'_k,1}^*, C_{i'_k,j'_k,2}' = C_{i'_k,j'_k,2}^*$. We prove this by showing that the probability of $C_{i'_k,j'_k,1}' \neq C_{i'_k,j'_k,1}^*, C_{i'_k,j'_k,2}' \neq C_{i'_k,j'_k,2}^*$ for some k is negligible.

According to Equations (2) and (4), the following equations hold

$$\begin{aligned} e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}')^{c_k}) &= e(g, \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}^*)^{c_k}) \\ \Rightarrow \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}')^{c_k} &= \prod_{k=1}^{\eta} (C_{i'_k,j'_k,1}^*)^{c_k}. \end{aligned}$$

Assume there exists a subset $L \subset [1, \eta]$ such that $\forall r \in L, C_{i'_r,j'_r,1}' \neq C_{i'_r,j'_r,1}^*$. Therefore, we remove the same items at both sides and get

$$\prod_{r\in L} (C_{i'_r,j'_r,1}')^{c_r} = \prod_{r\in L} (C_{i'_r,j'_r,1}^*)^{c_r}.$$

Set $C_{i'_r,j'_r,1}' = g^{\mu'_r}$ and $C_{i'_r,j'_r,1}^* = g^{\mu_r^*}$ for some unknown $\mu'_r, \mu_r^* \leftarrow \mathbb{Z}_p$ and $\mu'_r \neq \mu_r^*$, then we have

$$\prod_{r\in L} g^{c_r(\mu'_r - \mu_r^*)} = 1 \Rightarrow \sum_{r\in L} c_r(\mu'_r - \mu_r^*) = 0 (\text{mod } p).$$

Since c_r is unknown to $\mathcal{A}$, the probability of $C_{i'_r,j'_r,1}' \neq C_{i'_r,j'_r,1}^*$ is at most $1/p$, which is a negligible probability. Therefore, we have $\forall i, 1 \leq i \leq \eta, C_{i'_r,j'_r,1}' = C_{i'_r,j'_r,1}^*$. Similarly, $\forall i, 1 \leq i \leq \eta, C_{i'_r,j'_r,2}' = C_{i'_r,j'_r,2}^*$.

Hence, we have proved that $ct' = ct^*$ if $\mathbf{Verify}(ct, ct^*) \rightarrow 1$. That is, the attacker cannot generate an incorrect updated ciphertext while passing the verification. □

Table 2: Functionalities and features comparison of direct R-ABE

Schemes	Large universe	Ciphertext update	Security model	Verification
Attrapadung et al. [1]	×	–	standard model	×
Shi et al. [27]	×	anyone	random oracle model	√
Wang et al. [29] 1st scheme	×	–	standard model	×
Wang et al. [29] 2st scheme	√	–	standard model	×
Ours	√	anyone	standard model	√

Table 3: Efficiency comparison of direct R-ABE

Schemes	Size			Decryption cost	CTUpdate cost
	$\|pp\|$	$\|sk\|$	$\|ct\|$		
Attrapadung et al. [1]	$m+N+3$	$2(l+1)log(n)$	$\|S\|+1$ $+rlog(\frac{n}{r})$	$2(\|I\|+1)P$ $+(\|I\|+2)E$	–
Shi et al. [27]	$m+d+7$	$2l+2$	$2+\|S\|$ $+rlog(\frac{n}{r})$	$t_1P+\|I\|E$	t_2P
Wang et al. [29] 1st scheme	$2(m+2^d+1)$	$2l$	$2\|S\|+3$	$(l+2)P$ $+\|M\|E$	–
Wang et al. [29] 2st scheme	$2(m+2^d+1)$	$4l$	$2\|S\|+3$	$(2l+2)P$ $+\|M\|E$	–
Ours	6	$3l+2[log^2(n)$ $+log(n)]$	$2\|S\|+4r+1$	$(3\|I\|+4)P$ $+(\|I\|+1)E$	$4(r'+1)E$

7 Discussions

7.1 Performance Analysis

In this section, we compare the features and efficiency of our scheme with some existing direct R-ABE schemes [1, 27, 29]. This is shown in Table 2 and Table 3. Denote n to be the number of users in the system, m to be the maximum size of attributes, d to be the depth for all leaves in the full binary tree, l to be the size of rows in the LSSS matrix (or to be the size of the leaf nodes of access tree $\mathbb{A}$), I to be a subset of $\{1, 2, \cdots, l\}$, M to be a subset of the nodes in an access tree $\mathbb{A}$, r and r' to be the size of the revocation set R and R', respectively. N to be the maximum size of the cover set $cover(R)$. We denote

$$t_1 = 2|I| + d + 2 - depth(j),$$
$$t_2 = \sum_{j' \in cover(R')} (depth(j') - depth(j)).$$

E, P represent an exponentiation and pairing operation, respectively.

We first make a comparison in terms of functionalities and features in Table 2. Note that the second scheme presented by Wang et al. [29] and ours supporting large universe construction, but Wang et al.'s scheme [29] only achieved via fix a specific bound at stage of setup, and this approach places undesirable burden on the deployment of ABE schemes. Both Shi et al.'s scheme [27] and ours allow anyone to update ciphertexts and enable any auditor to verify whether the updated ciphertexts are correct. However, Shi et al.'s scheme [27] is secure in the random oracle model.

In addition, we make a analysis with respect to efficiency in Table 3. To the best of our knowledge, the storage overhead is mainly caused by storing the public parameters, the ciphertext and the secret key. It is easy to find that the sizes of the secret key and ciphertext in our scheme are close to the schemes [1, 27, 29]. However, the size of the public parameters is constant in our scheme. Meanwhile, due to the application of the SD method, our scheme could provide a smaller ciphertext component meant for performing revocation, at the cost of an admissible increased in the secret key size. On the other hand, although the efficiency of our scheme is lower than Attrapadung et al.'s scheme [1] and Wang et al.'s scheme [29] with reference to decryption, ours have more functionalities and features as analyzed above. In cihertext update, we use a trick similar to Shi et al.'s scheme [27], but our scheme is more efficiency: at most $4(r'+1)$ exponentiation operations. Whereas Shi et al.'s scheme [27] requires t_2 pairing operations.

7.2 Efficiency Improvement

As mentioned above, our scheme need to perform $3|I|+4$ bilinear pairing operations, which are considered the most expensive operation in pairing-based cryptographic protocols. These pairing operations lead to a heavy burden for the computation of decryption. Moreover, with the size of subset I increasing, the corresponding cost for the

computation of pairings would become higher. Therefore, it is indispensable to reduce the computation cost.

Outsourced ABE schemes might be the possible solution. Green et al. [11] first realized secure and efficient outsourcing of ABE decryption. Nowadays, many research works have been done on secure outsourced ABE schemes [13, 36]. Although outsourced ABE can reduce the computation cost of users, the systems have to introduce additional servers, which increases the system complexity [35].

Another possible solution is outsourcing computation techniques. Liu et al. [20] proposed a identity-based server-aided decryption scheme. Their scheme enables the receiver to decrypt the ciphertext without needing to compute paring with help of an external server. Based on the server-aided computation protocols, the notion of server-aided signature [7, 31] has been proposed in order to reduce the local computation. In 2014, Canard et al. [5] presented a secure and efficient delegating algorithm for pairing. This algorithm improved the computational complexity results. However, the solution was not feasible since exponentiation, membership test, and inversion were still required. Recently, Chen et al. [6] proposed an efficient and secure outsourcing algorithm for bilinear pairings in the two untrusted program model. In their scheme, the outsourcer never needs to perform any expensive operations such as exponentiations. It is not difficult to find that these secure outsourcing protocols and algorithms for bilinear pairings are also applicable to our scheme.

As a result, in our construction, we employ the secure outsourcing algorithm **Pair** presented by Chen et al. [6] to reduce the load of computation for the users. Similar to scheme [6], as a subroutine, the outsourcing algorithm **Pair** is invoked when users decrypt the ciphertexts. We show how the secure outsourcing algorithm **Pair** can be applied to our scheme. When a user decrypts a ciphertext ct, he runs the subroutine **Pair** to compute the message m (suppose the user identity $ID \notin R$ and the attribute set S satisfies the access structure (M, ρ)) as follows:

1) the user runs **Pair** to obtain $\mathbf{Pair}(C_0, k_{i,0}) \rightarrow \varphi_{i,0}$, $\mathbf{Pair}(C_{i,1}, k_{i,1}) \rightarrow \varphi_{i,1}, \mathbf{Pair}(C_{i,2}, k_{i,2}) \rightarrow \varphi_{i,2}$, $\mathbf{Pair}(C_0, D_{\tilde{i},\tilde{j},1}) \rightarrow \psi_1, \mathbf{Pair}(C_{i,j,1}, E_{\tilde{i},\tilde{j},1}) \rightarrow \psi_2$, $\mathbf{Pair}(C_0, D_{\tilde{i},\tilde{j},2}) \rightarrow \psi_3, \mathbf{Pair}(C_{i,j,2}, E_{\tilde{i},\tilde{j},2}) \rightarrow \psi_4$.

2) the user computes $B = \Pi_{i \in I}(\varphi_{i,0}\varphi_{i,1}\varphi_{i,2})^{\omega_i}$ and $B' = \psi_1\psi_2(\psi_3\psi_4)^{\frac{1}{(L_{\tilde{i}}||L_{\tilde{j}}) - (L_i||L_j)}}$.

3) the user computes $m = C/(BB')$ and outputs m.

Applying the secure outsourcing algorithm, the computation of decryption can be reduced to $|I| + 1$ exponentiation operations. This makes our system more suitable for practical applications.

8 Conclusions

In this paper, we have presented a directly revocable and verifiable key-policy attribute-based encryption (KP-ABE) scheme for large universe. Besides achieving efficient verification and large universe construction, the new scheme also utilizes the SD mechanism for direct revocation purpose. Compared with the CS scheme, the SD method greatly improves the broadcast efficiency and provides a smaller covering set, at the cost of an admissible increased in the secret key size. The DRV-KP-ABE scheme is selectively secure in the standard model.

Acknowledgments

We are grateful to the anonymous referees for their invaluable suggestions. This work is supported by the National Natural Science Foundation of China (Grants Nos. 61472470 and 61100229), and the Natural Science Basic Research Plan in Shaanxi Province of China (Grants Nos. 2014JM2-6091, 2015JQ1007 and 2015JQ6236).

References

[1] N. Attrapadung and H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes," in *Cryptography and Coding*, pp. 278–300, 2009.

[2] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Pairing-Based Cryptography (Pairing'09)*, pp. 248–265, 2009.

[3] A. Balu and K. Kuppusamy, "An expressive and provably secure ciphertext-policy attribute-based encryption," *Information Sciences*, vol. 276, pp. 354–362, 2014.

[4] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proceedings of the 15th ACM conference on Computer and communications security*, pp. 417–426, 2008.

[5] S. Canard, J. Devigne, and O. Sanders, "Delegating a pairing can be both secure and efficient," in *Applied Cryptography and Network Security*, pp. 549–565, 2014.

[6] X. Chen, W. Susilo, J. Li, D. S. Wong, J. Ma, S. Tang, and Q. Tang, "Efficient algorithms for secure outsourcing of bilinear pairings," *Theoretical Computer Science*, vol. 562, pp. 112–121, 2015.

[7] S. S. Chow, M. H. Au, and W. Susilo, "Server-aided signatures verification secure against collusion attack," *Information Security Technical Report*, vol. 17, no. 3, pp. 46–57, 2013.

[8] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *International Journal Network Security*, vol. 16, no. 1, pp. 1–13, 2014.

[9] P. Datta, R. Dutta, and S. Mukhopadhyay, "Fully secure unbounded revocable attribute-based encryption in prime order bilinear groups via subset difference method," *IACR Cryptology ePrint Archive*, vol. 2015, pp. 293, 2015.

[10] X. Fu, S. Zeng, and F. Li, "Blind expressive ciphertext policy attribute based encryption for fine grained access control on the encrypted data," *International Journal Network Security*, vol. 17, no. 6, pp. 661–671, Nov. 2015.

[11] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of attribute-based encryption ciphertexts," in *Proceedings of the 20th USENIX Conference on Security (SEC'11)*, pp. 34, 2011.

[12] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography (PKC'14)*, pp. 293–310, 2014.

[13] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[14] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal Network Security*, vol. 15, no. 4, pp. 231–240, 2013.

[15] K. Lee, "Self-updatable encryption with short public parameters and its extensions," *Designs, Codes and Cryptography*, vol. 79, pp. 121–161, 2016.

[16] K. Lee, D. H. Lee, and J. H. Park, "Efficient revocable identity-based encryption via subset difference methods," *IACR Cryptology ePrint Archive*, 2014.

[17] A. Lewko and B. Waters, "Unbounded hide identity-based encryption and attribute-based encryption," in *Advances in Cryptology (EUROCRYPT'11)*, pp. 547–567, 2011.

[18] K. Li and H. Ma, "Outsourcing decryption of multi-authority attribute-based encryption ciphertexts," *International Journal Network Security*, vol. 16, no. 4, pp. 286–294, 2014.

[19] Q. Li, H. Xiong, F. Zhang, and S. Zeng, "An expressive decentralizing key-policy attribute-based encryption scheme with constant-size ciphertext," *International Journal Network Security*, vol. 15, no. 3, pp. 161–170, 2013.

[20] J. K. Liu, C. K. Chu, and J. Zhou, "Identity-based server-aided decryption," in *Information Security and Privacy*, pp. 337–352, 2011.

[21] X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data," *International Journal Network Security*, vol. 16, no. 6, pp. 437–443, 2014.

[22] Z. Liu and D. S. Wong, "Practical attribute based encryption: Traitor tracing, revocation, and large universe," *IACR Cryptology ePrint Archive*, 2014.

[23] Y. Ming, L. Fan, H. Jing-Li, and W. Zhao-Li, "An efficient attribute-based encryption scheme with revocation for outsourced data sharing control," in *First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pp. 516–520, 2011.

[24] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proceedings of the 2013 ACM conference on Computer and Communications Security*, pp. 463–474, 2013.

[25] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology (CRYPTO'12)*, pp. 199–217, 2012.

[26] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology (EUROCRYPT'05)*, pp. 457–473, 2005.

[27] Y. Shi, Q. Zheng, J. Liu, and Z. Han, "Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation," *Information Sciences*, vol. 295, pp. 221–231, 2015.

[28] J. Singh, "Cyber-attacks in cloud computing: A case study," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 78–87, 2014.

[29] P. Wang, D. Feng, and L. Zhang, "Towards attribute revocation in key-policy attribute based encryption," in *Cryptology and Network Security*, pp. 272–291, 2011.

[30] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[31] W. Wu, Y. Mu, W. Susilo, and X. Huang, "Provably secure server-aided verification signatures," *Computers and Mathematics with Applications*, vol. 61, no. 7, pp. 1705–1723, 2011.

[32] M. Zareapoor, P. Shamsolmoali, and M. A. Alam, "Establishing safe cloud: Ensuring data security and performance evaluation," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 88–99, 2014.

[33] Y. Zhang, X. Chen, J. Li, H. Li, and F. Li, "Attribute-based data sharing with flexible and direct revocation in cloud computing," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 11, 2014.

[34] Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li, "Anonymous attribute-based encryption supporting efficient decryption test," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, pp. 511–516, 2013.

[35] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Efficient attribute-based data sharing in mobile clouds," *Pervasive and Mobile Computing*, 2014.

[36] Q. Zheng, S. Xu, and G. Ateniese, "Vabks: Verifiable attribute-based keyword search over outsourced encrypted data," in *Proceedings of IEEE INFOCOM*, pp. 522–530, 2014.

Hua Ma is a professor in the School of Mathematics and Statistics at Xidian University, Xian, China. Her research interests include design and analysis of fast public key cryptography, security theory and technology in electronic commerce, and network and information security.

Ting Peng is a master degree student in Mathematics at Xidian University. Her research focus on cryptography and network security.

Zhenhua Liu is an associate professor in the School of Mathematics and Statistics at Xidian University, Xian, China. His research interests include public key cryptography, cryptographic theory and security protocols in cloud computing.

An Improved DPA Attack on DES with Forth and Back Random Round Algorithm

Cai-Sen Chen[1], Xi Yu[2], Yang-Xia Xiang[2], Xiong Li[3], and Tengrun Li[4]
(Corresponding author: Cai-Sen Chen)

Ministry of Science Research, The Academy of Armored Forces Engineering[1]
No.21 Dujiakan Street, Fengtai District, Beijing City 100072, China,
Department of Information Engineering, Academy of Armored Force Engineering[2]
No.21 Dujiakan Street, Fengtai District, Beijing City 100072, China
School of Computer Science and Engineering, Hunan University of Science and Technology[3]
No.2 Taoyuan Road, Yuhu District, Xiangtan City, Hunan Province 411201, China
Kunming communication period, Kunming Railway Administration[4]
Kunming communication section, Kunming Railway Bureau, Yunnan province 650051, China
(Email: caisenchen@163.com)
(Received Jan. 07, 2016; revised and accepted Mar 10 & Apr. 9, 2016)

Abstract

The power leakage problems of smart card chip during the process of DES encryption are analyzed, we propose two attack algorithms on DES with forth and back random round algorithm respectively, include the accumulative attack algorithm and segmented attack algorithm. We provided an improved analysis algorithm based on the segmented attack by using a new correctional factor: the cliffy characteristic of the peak value. Finally, the first round key of DES was recovered successfully, and the DES key was deduced. It proves that the countermeasure with forth and back random round cannot t protect the security of smart card. We implement the DPA attack on DES with forth and back random round algorithm in the mathlab simulation experiment environment, the experiment results show that the feasibility and effectiveness of DPA can be improved by the advanced algorithm.

Keywords: Accumulative attack algorithm, DES algorithm, differential power analysis (DPA) attack, forth and back random round, segmented attack

1 Introduction

The traditional cryptanalysis methods are invalid with the improvement of the cryptographic algorithms and the increases of key length, a new cryptanalyst analysis algorithm was proposed based on the leakages information during the process of cryptographic chip, such as execution time, power consumption, etc. This attack algorithm was defined as Side Channel Analysis (SCA), which is a new direction of the cryptanalysis. Due to the features of chip circuit, there are side channel leakages during the process of encryption for smart card. The experiment result had proved that these leakages were closely related to the data and the key during the process of the cryptographic execution. As the high facility of implementation, low consumption, Side Channel Attack has become a new hotspot in the research field of cryptanalysis [7, 12, 14].

Paul Kocher proposed Power Analysis (PA) attack [10] in 1998, and gave three methods of it at the same time, Simple Power Analysis (SPA), Differential Power Analysis (DPA) and Correlative Power Analysis (CPA) [9, 17]. Due to its low consumption, fast speed, Power Analysis attack can collect the power consumption of the smart card to crack the key without breaking the chip, which could cause a large threat to safety of the smart card. SPA attack cracks the key using one or a few power consumption curve directly or indirectly, without a large number of statistical analyses. While DPA attack requires collecting a large number of power consumption curve, the attackers do not need to know the detail information of the target device, they only need to know which cryptographic algorithm was performed, so this attack has strong resistance to the noise [5].

Recently, study of DPA attack on smart card had achieved some progress. Literature [13] proposed a method to analyze power consumption using Neural Network, which combines the advantages of SPA and DPA, and even could restore the key with only one power consumption curve. Literature [4] provided a series statistical test methods of DPA attack. Literature [3] built the basic power simulation model for the first time, and literature [1] applied the basic power simulation model to smart card chip. Nowadays, based on the model, more effective

models have been proposed [6, 15, 16]. Brier presented a method of correlation coefficient which has become a new direction of the DPA study [2].

In order to improve the safety of the chip to prevent DPA attack, literature [11] put forward a countermeasure method for linear and nonlinear part during the process of DES encryption, which could randomize the intermediate values based on the software programming of MASK [18], and make the power consumption and intermediate values are not correlative, so it can prevent the power attack, which has become a hot area of DPA countermeasure research. However, the small amount of MASK has not sufficient randomness, and it requires long operation time and large power consumption, then researchers presented a random round model implemented in hardware. Radom round concealed the real interval [17] of DES encryption with pseudo random input/output and round number. Dummy round resistant countermeasures on the basis of the random round position can be divided into two kinds, forth-back dummy round and middle dummy round. This article mainly focuses on the study of DPA attack against DES algorithm with forth-back dummy round [8].

2 DES and DPA Model

2.1 DES Algorithm

DES faces a security challenge as its fixed and short key, but due to its fast speed and low consumption, nowadays most of smart cards still use the DES as a security method. Each round in 16 rounds of DES there are 8 looking up S-box table operation. The input of S-box is a 6-bit subkey XOR with a 6-bit R register value, and the output of S-box is a 4-bit data. And then the 32 bit output will be restructured, XOR with the value of L register, then L and R register value changed each other. Each round deals with data in the same way. Figure 1 shows the first round encryption process.

Put 64 bit input plaintext into left part L_1(32 bit) and right part R_1(32 bit), in which R_1 will be 48 bit after expansion, and then get 32 bit through S box after R_1 XOR with subkey Key_0. The 32 bit XOR with L_1 becomes the R_2, while this round R1 become next round.

$$L_2 = R_1 \quad (1)$$

$$R_2 = L_1 \oplus f(R_1, Key_0). \quad (2)$$

Function f indicates these steps of plaintext expansion, exclusive, data compression and displacement, which is nonlinear operation steps in the process of encryption.

2.2 Analysis Model of Power Consumption

In power analysis attacks, the attacker needs to make a relationship between the data and the power consumption of the attacked device. In the practical implementation, they are not a direct linear relationship; attacker only needs to know the relative value of power consumption instead of the absolute value. Commonly we use the Hamming-distance model (HDM) and Hamming weight model (HWM).

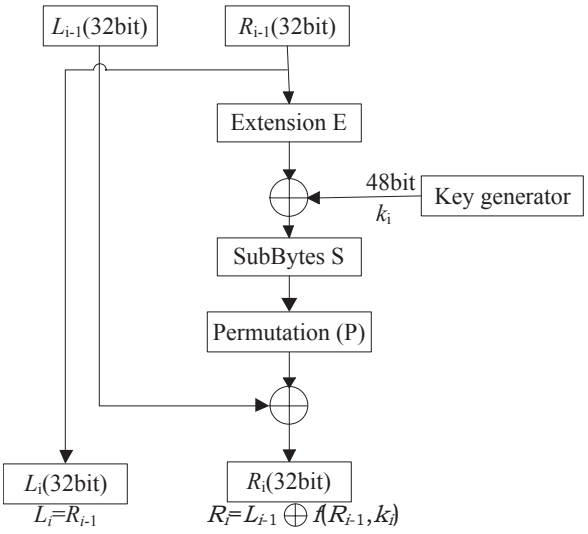


Figure 1: The i_{th} round of DES encryption process

A device of digital circuit may make a flip in a moment, it will produce dynamic power consumption (replaced with power in the follow), and thus the more device flips, the more power consumption. Therefore, the number of device flips can be relative with the relative value of the power consumption, which is the basic principle of HDM. HDM generally can be used to describe the power consumption of the bus and register.Attackers will get the corresponding information through establishing a linear relationship between the number of device flips from timing start to end and power consumption using HDM. Theoretical studies show that power consumption of device flip before and after is linear correlative with the hamming distance of binary data before and after.

$$P = k * HD(V_0, V_1) + d. \quad (3)$$

k and d is the constant decided by the device characteristics.

From the point of SPA, it obtains the hamming distance information, detects the size of relative power consumption. From the point of DPA, it takes hamming distance as a function to attack the power curve. The biggest drawback of HDM is that the attacker must know the initial value and the final state to calculate the correct hamming distance, but in reality it is not necessarily to get these two values.

HWM is much simpler than HDM. In HWM, attackers assume that the size of power consumption is proportional to the number of the bit 1 in the processing value V_1, which is also the HW of value V_1.

$$P = k * HD(V_1) + d. \quad (4)$$

k and d are the constant values which are decided by the device characteristics.

The advantage of HWM is that it doesnt need to know all the status value of circuit unit, and just needs the hamming weight of a certain status, to construct the corresponding relationship between power curve and value. All experimental results in this article are based on HWM.

2.3 Principle of Differential Power Attacks

Nowadays most of the integrated circuits are composed of CMOS, the energy consumption of CMOS can be divided into two parts: static and dynamic energy consumption. The former refers to the components without any data processing, namely the circuit without own energy consumption of gate flip; the latter refers to the energy consumption of gate flip, in addition to static energy, when internal element or output signals changes.

In the common CMOS circuit, the total energy consumption of the circuit is mainly caused by dynamic energy consumption, and the dynamic energy consumption is mainly related to the data processing. According to the type of energy consumption described before, it can be seen that the dynamic energy consumption is associated with the change of the data, which is a large proportion of the total energy consumption in the circuit, so the energy consumption of the circuit mainly depends on the data processing, it is also the physical circuit basis of energy attacks.

The DPA attack mainly includes two steps [9]: waveform acquisition and data analysis. Power waveform acquisition is on the hardware part, this article mainly use the simulation software, so it mainly relates to data analysis, the detailed steps are as follows:

1) Encrypt different plaintexts N groups and measure the power consumption curve in the first round of the DES operation process $\{P_i|1 \le i \le N\}$.

2) Select a bit of the output value of the first S box operation in the first round as a function D, with b indicates the bit value, called intermediate value. It is easy to know that all depends on the key K and plaintext M to b, which can be indicated as D(K, M). Make a guess on related value when attacking to get the corresponding intermediate value. According to the intermediate value, N groups power consumption curve can be divided into two categories:

$$S_1 = \{P_1|D = 1, 1 \le i \le N\} \quad (5)$$
$$S_0 = \{D = 0, 1 \le i \le N\} \quad (6)$$

3) The average power consumption of set S_1 and S_0 calculated separately:

$$A_1 = \frac{1}{S_1}\sum S_1(i) \quad (7)$$
$$A_0 = \frac{1}{S_0}\sum S_0(i) \quad (8)$$

Among them, S_1 and S_0 indicates the number of power consumption curves corresponding collection:

$$|S_1| + |S_0| = N. \quad (9)$$

4) The difference value between them is:

$$T = A_1 - A_0. \quad (10)$$

If the key is guessed correctly, and then the classification of the power consumption curves is right, that is to say, all the curves of $b = 1$ points to S_1, and the rest curves of $b = 0$ are assigned to S_0, so there is an obvious peak in power consumption curves. If the key is not right, the peak of T will be very little or no.

Classifying collection equate with classifying collection using a random function. As the mathematical statistics theory, when using a random function divides collection into two, elements in the two collections tend to infinity; the difference of average between the two subsets will tend to 0. If the sample space of power consumption curves analysis is big enough, the chosen model appropriates and the effects of sampling noise is small enough, we will get the difference power consumption curves with the peak after the above difference statistical method, the position of the peak is the guessed key. According to the above method calculated, we can obtain the key corresponding to the first S box and repeating to acquire the key corresponding to rest S boxes. Finally, we will obtain the 48 bit key actually used in this way and guess the rest 8 bit in exhaustive method to acquire the complete 56 bit key. Such workload is $2^6 \times 8 + 2^8 = 768 \le 256 = 7.2 \times 10^{16}$. Detailed algorithm is shown in Figure 2.

3 Differential Power Analysis of DES Implemented With Forth and Back Random Round Algorithm

3.1 DES Implemented with Forth and Back Random Round Algorithm

Generally, the principle of DPA attacks on smart card DES chip is aligning to find the starting position of DES data encryption by filtering the power wave, then cracking the key of the first or 16^{th} round of DES encryption. Based this principle, C. Herbst puts forward a defense algorithm using random round, namely before or after or in the middle of the DES encryption joining a random pseudo rounds, whose rules and operations are the same, but each round input is not plaintext instead a random number, while the real DES encryption still encrypt plaintext make the normal output.

Due to the number of random round in DES encryption becomes more and more, the ability of resistance to the DPA attacks becomes also stronger. However the more

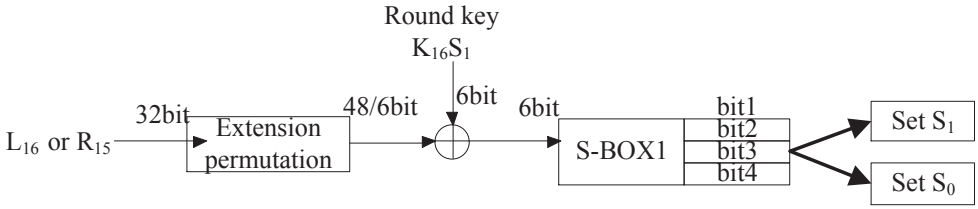


Figure 2: Distinguish between power consumption function

the number of random round and the more widely distribution is, the slower speed of chip CPU is, some smart card used forth and back random round model are shown as in Figure 3.

Figure 3 shows the relationship between power consumption when plaintext encrypted and random round. The number of random round each encrypted is not fixed and the real encryption interval of DES algorithm is uncertain, so even if attackers have got the power consumption curves, they also don't know the position of the real encryption interval.

3.2 Two Kinds of DPA Analysis Attacks Against DES Algorithm

3.2.1 Accumulative Attack Algorithm

Figure 4 shows the average cumulative power consumption of the previous round (DR + 1) before the accumulative attack algorithm. This method can be regarded as a derivation of alignment operation: the average power consumption as the first round and make it contains the power consumption of the first round or even complete real DES algorithm indirectly. This method can guarantee that attack contains useful information in the first round, just with bigger noise which generated from random round.

According to the principle shown in Figure 4, we will put forward a concrete algorithm.

In the accumulative attack algorithm, we accumulate the former (DR+1) power wave, then average, and treat it as the new first round of DES algorithm. Because the maximum number of dummy rounds is DR, the front (DR+1) rounds include at least the first round of DES. The new first round contains real first round and noise due to other rounds, so we attack the new round to get the key by using DPA.

3.2.2 Segmented Attack Algorithm

Compared with the accumulative attack algorithm, nodes and methods of segmented attack algorithm are different. Algorithm 1 chooses the former (DR+1) rounds and add power consumption together to get the average power consumption, as the first round of the DES, while Algorithm 2 does not accumulate power wave, but to attack each former (DR+1) rounds respectively. As shown in Figure 5.

Algorithm 1 Accumulative attack algorithm

```
Begin
input: M; //the set of plaintext, each plaintext indicated as M_i, i={1,..., size(M)};
Wave; //the set of power consumption curves, each curves indicated as Wave_i, i=1,..., size(M), corresponding to plaintext;
DR; // the maximum number of random round;
DRinput; //input of each random round;
output: number; //the possibility sort of key corresponding to 8 S-box;
Subkey(i); //the subkey corresponding to each S-box
while j < size(M,1) do
  while i < (DR+1) do
    Temp=Temp+Wave(i) //a loop of waveform accumulation
    Wave(j)=Temp/(DR+1) //make it as "the first round" after accumulation
  end while
end while
DPA algorithm
number;
Subkey(i);
End
```

According to the principle in Figure 5, we will put forward concrete algorithm.

The basic idea of Algorithm 2 is that: from left to right, we regard each round as imaginary "the first round" of DES, and get each subkeys of them by using DPA. Each round of the same S-box has subkey possibility sorting by order from high to low, and store the top 10 subkeys in the set $KeyRank(1)$; we put all $KeyRank$ sets of the same S-box together and find the key with highest occurrence probability and top ranking, and then the subkey is the correct key of the S box. Repeat the process for each S box, we will find all the right subkeys.

4 The Simulation Experiment Results And Complexity Analysis

4.1 Simulation Experiments

Due to the laboratory conditions limited, this article makes simulation experiments on matlab to verify the re-

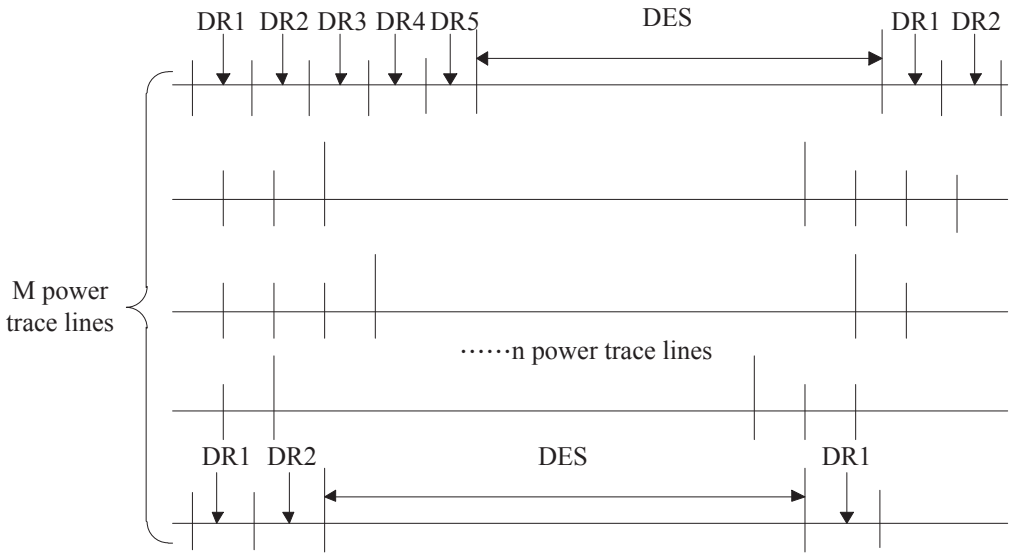


Figure 3: Forth and back random round model

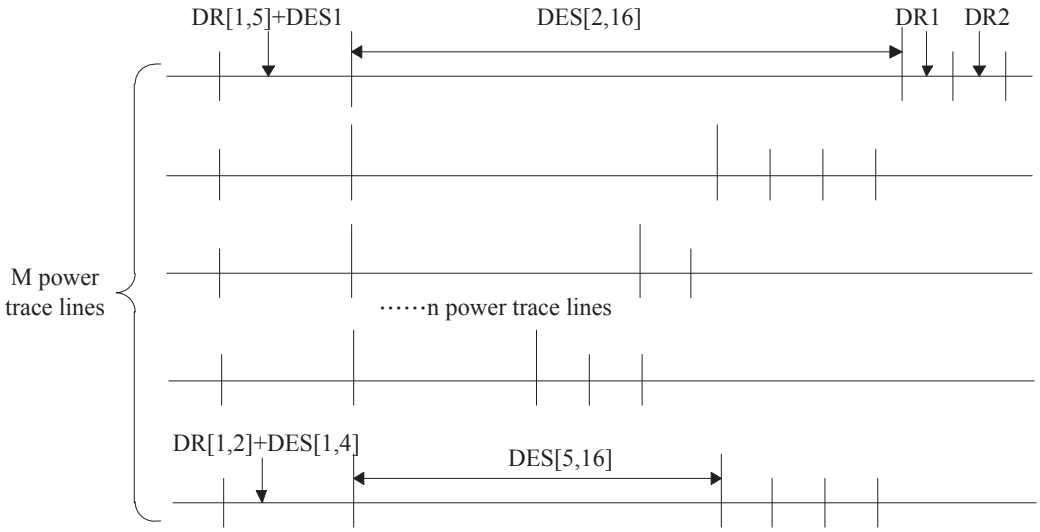


Figure 4: The principle of the accumulative attack algorithm

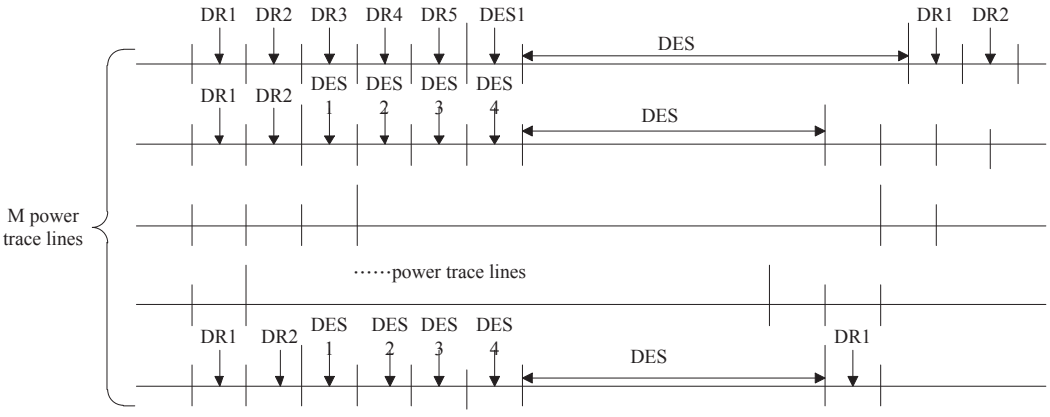


Figure 5: The principle of the segmented attack algorithm

Algorithm 2 Segmented attack algorithm

1: Begin
input: M; //the set of plaintext, each plaintext indicated as M_i, i=1,..., size(M);
Wave; //the set of power consumption curves - each curves indicated as $Wave_i$, i=1, ..., size(M), corresponding to plaintext;
DR; // the maximum number of random round;
DRinput; //input of each random round;
output: KeyRank(i) // the possibility sort of key corresponding to 8 S-boxi=1, ..., DR, DR+1, the size is 8*64;
Number; // the possibility sort of (DR+1) corresponding to 8 S-box, the size is (DR+1)*64
2: **while** $i < (DR + 1)$ **do**
3: **while** $j < size(M, 1)$ **do**
4: DPA algorithm
5: KeyRank (j);
6: **end while**
7: **end while**
8: Number(i)=KeyRank(j)(i,:);// extract (DR + 1) KeyRank (j) of the same line Number as all the lines of the Number (I)
9: End

search content. As already stated, the study results of Hamming Weight Model have shown that the results of power consumption sampling in simulation experiments are basically identical with the actual. The target of experiment is the first round of DES keys, setting the maximum random number of previous round is 5 and the length of key is 64.

4.2 The Results of the Experiment

Based on the accumulative attack algorithm and the segmented attack algorithm above mentioned, we make the simulation experiment on MATLAB and put forward an improved algorithm based on the results of segmented attack algorithm.

4.2.1 The Results of Accumulative Attack Algorithm

According to accumulation attack algorithm introduced in Algorithm 1, we design programming ideas to acquire subkey and its possibility sort corresponding to each S-box using MATLAB, and the most likely key is the right one.

Table 1 shows the possibility matrix (size is 8 * 64) of key corresponding to 8 S-boxes. The probability is derived from the height of peak in the DPA attacks corresponding to each subkey, and t the larger the peak value is, the higher the possibility of it. Each row of the matrix represents the 64 guessing keys corresponding to S-box, ordered from big to small of the peak value, and the first column represents the key with the highest peak values.

Table 1: the possibility sort of subkey corresponding to 8 S-boxes based on accumulation attack algorithm

	subkey								
Sbox	1	2	3	4	5	6	7	...	64
1	29	47	24	4	35	8	28	...	52
2	48	46	37	45	47	20	27	...	55
3	39	35	10	37	43	41	42	...	22
4	55	26	10	20	34	42	60	...	25
5	39	52	1	36	60	14	48	...	42
6	44	16	23	57	48	48	51	...	37
7	16	21	54	22	28	28	48	...	27
8	14	26	38	10	25	25	29	...	2

Experiments show that when the number of power consumption track become bigger and bigger, for the same key, the number of the first column is more stable, which is the right sort of subkey remains about the same.

Table 2: The success rate of accumulative attack algorithm

Power Trace Line Number	Correct Key Number	Incorrect Key Number	Success Rate(%)
50	1	7	12.5
80	1	7	12.5
100	3	5	37.5
200	4	4	50
300	6	2	75
400	6	2	75
700	7	1	87.5
1000	7	1	87.5
1300	8	0	100
2000	8	0	100
3000	8	0	100

Table 2 statistics that the success rate of the DPA attacks to DES algorithm using accumulative attack algorithm. It is shown that we can break the DES algorithm by 1300 power consumption curves of key at most in accumulative attack algorithm.

4.3 The Results of Segmented Attack Algorithm

According to Algorithm 2, we make the simulation experiment on MATLAB and get the key corresponding to each S-box.

The maximum number of random round is 5, the round of segmented attack algorithm is 6, and the results of each round as a row of Table 3 shown, in addition, each column is sorted by the possibility (the larger the peak value is, the higher the possibility of it)of the subkey. Selected the

Table 3: The guessing key sort of S2-box

	subkey								
Round	1	2	3	4	5	6	7	...	64
1	11	49	3	2	53	57	26	...	15
2	37	27	45	4	20	63	39	...	56
3	37	48	46	47	38	51	63	...	1
4	37	4	47	19	48	28	46	...	20
5	37	48	46	27	39	56	11	...	17
6	37	4	50	27	45	3	38	...	19

top 10 columns, we make a statistical about occurrences of each guessing subkey with KeyRank, and the results shown in Figure 6.

As Figure 4.1 shown, the subkey corresponding to S-box is 37. According to the method cycled, we can obtain the key corresponding to 8 S-boxes. Table 4 shows the success rate of segmented attack algorithm.

4.4 An Improved Analysis Algorithm with an Introduced Modifying Factor

Compared with Table 2 and Table 4, we can see that when the success rate of attack reaches 100%, the used sample size of segmented attack algorithm is larger than accumulative attack algorithm. In principle, due to average all power consumption curves in accumulative attack algorithm, the noise will be increased anyway, while segmented attack algorithm is not, it is piecewise attack, only compared occurrences and ranking position of each period key to find the correct one in the end, which means Algorithm 1 with high noise while Algorithm 2 with large workload. However, theoretically, Algorithm 2 needs less power consumption curves, but in fact the opposite is true. After repeated experiments, we found that the right key not only depends on the ranking position but also the frequency at the same time in Algorithm 2, and sometimes these two factors are conflicts with each other, so it is too difficult to find a balance.

Verified by the experiment, when a number of guessing subkeys occurred at the same time, the probability of that one of these same frequency subkeys nearby is smaller, which probably to be the correct subkey. Therefore, we should join a correction factor, the steepness of peak, which means occurrences of the right key nearby is few.

As shown in Figure 7, the occurrences of the location corresponding to subkey 37, 46, 47, achieve the maximum. Identify the right values, we should take the steepness of peak into consideration. From the graph, compared with other two numbers, the frequency nearby key 37 is much smaller than itself, so the location of key 37 is the correct key.

Added random round, we measure 3000 power consumption curves corresponding to 100 random keys, calculated according to the Algorithms 1 and 2 respectively, and join the steepness of peak, to get the success rate of these two algorithm in different circumstances using true and false subkey (see Table 5).

In conclusion, two algorithms crack the key successfully, just the workload different. Algorithm 1, by contrast, the calculation of time and effort is less than the Algorithm 2, but easily affected by noise, so the Algorithm 1 is suitable for the situation which accuracy is not high but short time requirements of random, while Algorithm 2 can be applied to the situations which is high precision of key crack regardless of the time cost.

5 Conclusions

In this article, in view of the dummy round defensive measures, we make study on the DPA attack with DES algorithm of smart card. Firstly, according to the dummy round defensive feature, we present two kinds of DPA attack, accumulative attack and segmented attack algorithm; Secondly, we apply these two algorithms respectively to simulation experiments for smart card in MATLAB, which crack the key successfully and give the attack rate of both; Finally, introduced the modifying factor, we acquire the improved algorithm against segmented attack algorithm, in order to reduce the number of power consumption curves required for the attack.

The analysis of experimental results shows that the two kinds of attack algorithm is feasible, in addition, the running time and accuracy of each algorithm determines the scope of application. Next, based on the existing research, we will make efforts to further expand the design for the DPA attack algorithm using completely dummy round, and provides reference for a smart card password cracking.

Acknowledgments

The authors would like to thank anonymous reviewers for their valuable comments. This research was supported by the National Natural Science Foundation of China under Grant No. 61402528. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. L. Akkar, R. Bevan, P. Dischamp, and D. Moyart, "Power analysis, what is now possible?" *Lecture Notes in Computer Science*, vol. 1976, pp. 489–502, Springer, 2000.

[2] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," *Lecture Notes in Computer Science*, vol. 37, no. 22, pp. 8004–8010, Springer, 1998.

[3] S. Chari, "A cautionary note regarding evaluation of aes candidates on smart-cards," in *Advanced Encryp-*

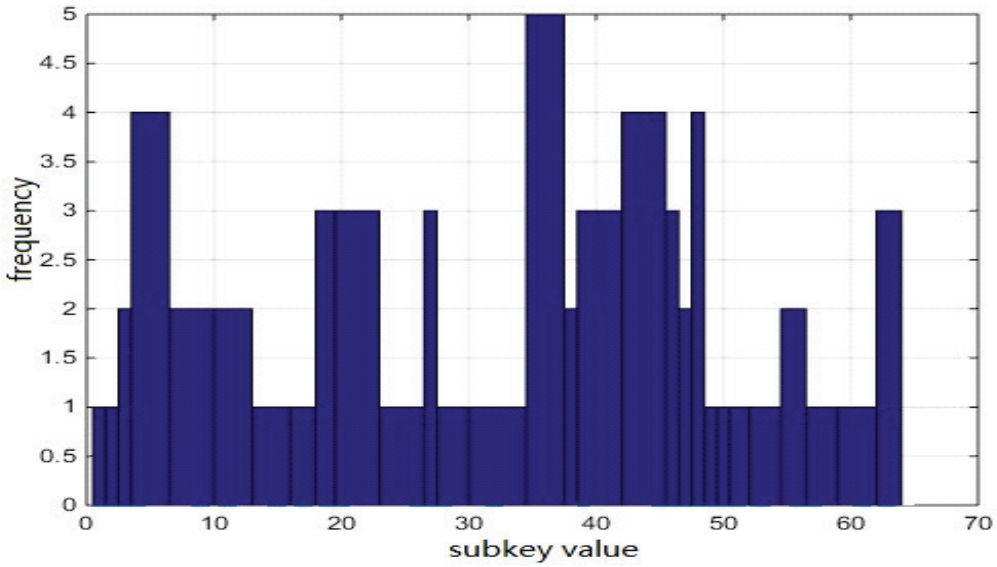


Figure 6: The key frequency of S2-box

Table 4: The success rate of segmented attack algorithm

power trace line number	correct key number	incorrect key number	success rate(%)
80	1	7	12.5
100	1	7	12.5
200	2	6	25
300	3	5	37.5
500	4	4	50
700	5	3	62.5
1000	6	2	75
1500	7	1	87.5
2000	8	0	100
3000	8	0	100
4000	8	0	100
5000	8	0	100

Table 5: Analysis results of success rate for two algorithms

	true: $false > 8:0$	**true:** $false > 7:1$
Accumulative attack algorithm	93	100
Segmented attack algorithm	95	99

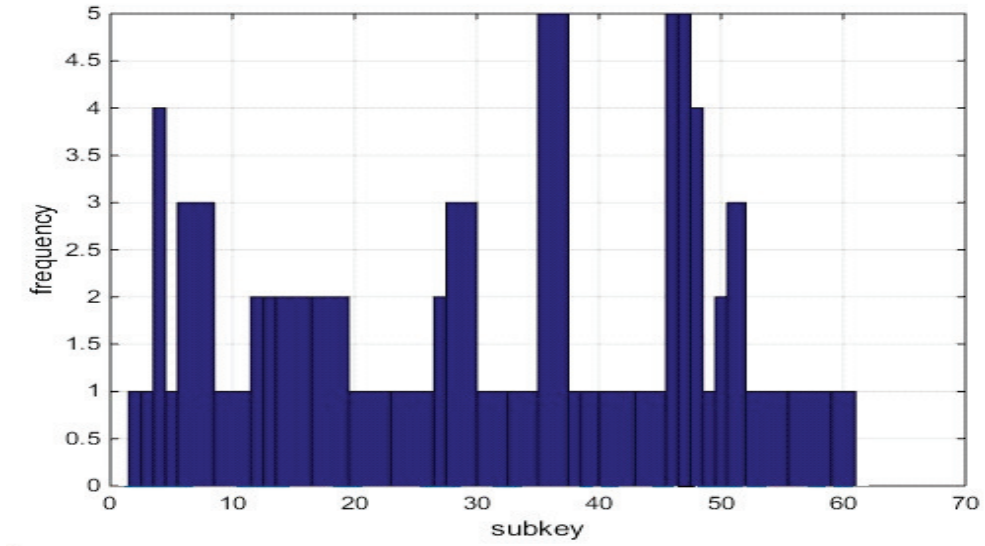


Figure 7: The frequency statistics of key corresponding to S2-box

tion Standard Candidate Conference, pp. 133–147, 1999.

[4] J. S. Coron, P. Kocher, and D. Naccache, *Statistics and Secret Leakage*, Springer Berlin Heidelberg, 2015.

[5] D. L. Delivasilis and S. K. Katsikas, "Side channel analysis on biometric-based key generation algorithms on resource constrained devices," *International Journal of Network Security*, vol. 3, no. 1, pp. 44–50, 2006.

[6] Y. Fei, Q. Luo, and A. A. Ding, *A Statistical Model for DPA with Novel Algorithmic Confusion Analysis*, Springer Berlin Heidelberg, 2012.

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

[8] C. Herbst, E. Oswald, and S. Mangard, "An aes smart card implementation resistant to power analysis attacks," in *Applied Cryptography and Network Security, Second International Conference (ACNS'06)*, pp. 194–206, 2006.

[9] L. I. Jing and L. I. Lin-Sen, "Differential power analysis method for des encryption in IC card chip," *Computer Engineering*, vol. 39, no. 7, pp. 200–204, 2013.

[10] P. C. Kocher, J. M. Jaffe, and B. C. Jun, "Differential power analysis," in *Advances in Cryptology (CRYPTO'99)*, LNCS 1666, pp. 388–397, Springer, 1999.

[11] S. Mangard, E. Oswald, T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer, 2008.

[12] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[13] Z. Martinasek and V. Zeman, "Innovative method of the power analysis," *Radioengineering*, vol. 22, no. 2, pp. 586–594, 2013.

[14] A. Mersaid, T. Gulom, "The encryption algorithm AES-RFWKIDEA32-1 based on network RFWKIDEA32-1," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 1–11, 2016.

[15] A. Moradi, M. Salmasizadeh, M. T. M. Shalmani, and T. Eisenbarth, "Vulnerability modeling of cryptographic hardware to power analysis attacks," *Integration the VLSI Journal*, vol. 42, no. 4, pp. 468–478, 2009.

[16] E. Peeters, F. X. Standaert, and J. J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integration the VLSI Journal*, vol. 40, no. 1, pp. 52–60, 2007.

[17] F. Yan, *The Study of Attacks and Defenses Based on AES Algorithm*, Master's Thesis, Beijing Jiaotong University, 2013.

[18] Z. Zhuang, J. Chen, and H. Zhang, "A countermeasure for DES with both rotating masks and secured S-boxes," in *Tenth International Conference on Computational Intelligence and Security*, pp. 410–414, 2014.

Caisen Chen was born in 1983. He received his master's degree in computer software and theory and a Ph.D. degree in network security technology from Ordnance Engineering College in 2009 and 2011, respectively. Dr.

Chen now is a lecturer of Academy of Armored Force Engineering in Beijing, and his current research interests include information security and implementation attack analysis on Cryptosystems. He also researches the security of mobile device.

Xi Yu was born in 1992. She received her B.S. degree in information security from Sichuan University in 2014. She is currently a M.S. student in Department of Information Engineering, Academy of Armored Forces Engineering, Beijing, China. Her main research interest includes information security and side channel analysis of block ciphers. Email: yuxijxx@163.com.

YangXia Xiang was born in China in 1981.She received the Diploma in Computer Science and Technology from the University of Air Force Engineering,China, M.Sc in computer software architecture from Ordnance Engineering College,China.She now is a lecture at the Department of Information Engineering in Academy of Armed Force Engineering. Her current research interests include computer system and net work security.

Xiong Li was born in China in 1984. He received his master's degree in mathematics and cryptography from Shaanxi Normal University (SNNU), China in 2009 and Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT), China in 2012. Dr. Li now is a lecturer at School of Computer Science and Engineering of the Hunan University of Science and Technology (HNUST), China. He has published more than 40 referred journal papers in his research interests, which include cryptography and information security, etc. He has served on TPC member of several international conferences on information security and is a reviewer for more than 20 ISI indexed journals. He is a winner of the 2015 Journal of Network and Computer Applications Best Research Paper Award.

Tengrun Li was born in China in 1990. She holds a B.Eng. in Communication Engineering, an M.Eng. in Communication and Information Systems, from the Beijing Jiaotong University. Her current research interests include bypass attack and defense.

SecureCoin: A Robust Secure and Efficient Protocol for Anonymous Bitcoin Ecosystem

Maged Hamada Ibrahim

(Corresponding author: Maged Hamada Ibrahim)

Department of Electronics, Communications and Computers, Faculty of Engineering, Helwan University
No. 1, Sherif St., Helwan, P.O. 11792, Cairo, Egypt
(Email: mhii72@gmail.com, maged_ismail@h-eng.helwan.edu.eg)

(Received Jan. 19, 2016; revised and accepted Apr. 7 & Apr. 25, 2016)

Abstract

Bitcoin is the first decentralized peer-to-peer electronic virtual asset and payment cryptocurrency, through which, users can transact digital currency directly, without the need for an intermediary (or authority), using a hashed version of cryptographic public keys, as pseudonyms called addresses. The Bitcoin ecosystem was supposed to be anonymous and untraceable. However, transactions from input to output addresses of the Bitcoin users are observed to be linkable, therefore, missing unlinkability as an important requirement of anonymity. Several protocols appeared to enhance Bitcoin users' anonymity and to ensure unlinkability of input-output addresses, to make input and output addresses of transactions unlinkable to each other, and hence untraceable. In this paper, we spot several vulnerabilities in the most recently proposed protocols, then we propose SecureCoin as an efficient protocol for anonymous and unlinkable Bitcoin transactions that covers these vulnerabilities in a robust and secure way and in full compatibility with the standard Bitcoin ecosystem. Our protocol provides better protection for the participating peers against malicious behavior of minority of the peers and protection against the most serious sabotage attack attempted by any number of saboteur peers. We analyze the security properties of our protocol and evaluate its efficiency. Finally, we compare the performance of our protocol with the recently proposed protocols and show that our protocol is computationally efficient and requires less Bitcoin fees.

Keywords: Anonymity, bitcoin, cryptocurrency, ECDSA, oblivious shuffling, silk road, unlinkability

1 Introduction

Bitcoin is a decentralized electronic currency protocol, that realizes worldwide of fast peer-to-peer transactions, and consequently payments with low, or near zero, transaction processing charges. In order to avoid the dependency on a central trusted authority charged with the issuance and control of currency, Bitcoin functions through a peer-to-peer topology. In this ecosystem, it is not possible to manipulate the value of the digital coins (bitcoins) or produce inflation through the overproduction of currency. Transactions and the creation of bitcoins are managed by the network itself; the creation of digital coins in a controlled and decentralized manner through the process known as mining. Cryptographic primitives guarantee the security of transactions. Coins can only be spent by the owner of their pseudonyms, and they can only be used in a single transaction with no chance of duplication. The supervision and management institutions that operate in traditional trusted centralized systems do not exist for Bitcoin [2].

Of the participants in the Bitcoin system, two – not necessarily mutually exclusive – groups can be distinguished:

Normal users. Users of the Bitcoin system who buy or sell for goods and services with coins, producing transactions in the system.

Miners. Special users who dedicate computing power to verify new transactions, creating what are known as transaction blocks. The calculations required to do this are expensive in computing power, which is why these users are rewarded.

1.1 Bitcoin Components and Processes

Bitcoin addresses. This is a user's digital address, also called pseudonym, which contains coins and which is used to make and receive transactions, similar to a bank account. A given user may have as many addresses as he wishes, and they are identified by a public key. Bitcoin uses the ECDSA (Elliptic Curve Digital Signature Algorithm) to sign its transactions, using parameters recommended by the Standards for Efficient Cryptography Group (SECG), secp256k1 [5]. The signatures use DER encoding.

ECDSA offers many advantages over other digital signature systems (e.g. RSA and DSS) which make it ideal to be used for a distributed Internet protocol. ECDSA provides relatively short keys and signature lengths, and a faster generation and verification. On being identified by the ECDSA public key, all the operations carried out with this address have to be supported by the use of the corresponding private key. The holder of the private key is the owner of the coins associated with the corresponding address.

Wallets. Personal virtual storage, similar to a physical/pocket wallet, where users' coin addresses and the payments made with them are stored and managed. Wallets are thus a grouping pairs of public and private keys and are used to carry out other tasks, for example, preparing transactions.

Transactions. A transaction is the transfer of coins from Bitcoin input address I to another output/destination address D. To create a transaction, the owner of address I signs a transcription for address D (amongst other data) with the private key associated with address I, so that the whole network knows that the new legitimate owner is the owner of address D. There are three main types of Bitcoin transaction forms [2] that will be used in our SecureCoin protocol:

- One-to-one form. A commonly used form of transaction is a simple payment from one address to another, which often includes some change returned to the original owner. This type of transaction has one input and one output if no change is returned, or in general, one input and two outputs.
- Aggregation form. Another common form of transaction is a transaction that aggregates several inputs into a single output. This represents the real-world equivalent of exchanging a pile of coins and currency notes for a single larger note. Transactions like these are frequently generated by wallet applications to clean up lots of smaller amounts that were received as change for payments, by transferring them to a new fresh address owned by the same user.
- Distribution form. Another transaction form that is seen frequently on the Bitcoin ledger is a transaction that distributes one input to multiple outputs representing multiple recipients. This type of transaction is sometimes used by commercial entities to distribute funds, such as when processing payroll payments to multiple employees.

Blocks. This is a structure that aggregates transactions. Transactions whose confirmation is pending are aggregated together as a block in a process which is known as mining.

Blockchain (Ledger). This is the public record of verified Bitcoin transactions in chronological order. When a block has been checked and confirmed, through mining, it is included as part of the chain.

Cryptographic hashes. In the hash calculations carried out in Bitcoin, the SHA-256 standards are used, and when shorter hashes are required, RIPEMD-160. Normally the hash calculations are carried out in two phases: the first with SHA-256, and the second, depending on the length desired, with SHA-256 or RIPEMD-160.

Random numbers and nonces. In Bitcoin, random numbers and nonces are used directly in the formation and generation of blocks. To make a new block, a random number that satisfies certain requirements needs to be found. Random numbers are also indirectly used in Bitcoin as part of the digital signature algorithm (ECDSA).

Proofs-of-work. Proofs-of-work are the main component guaranteeing the legitimate behavior of Bitcoin network. The idea "in brief" is that, verifying/calculating new transaction blocks ensures a high computational cost, such that to control the network and to regulate the rate at which new coins are generated. This control of complexity in the calculation of new blocks is carried out by requiring that the hash for each new block starts with a given number of zeros. This required number of zeros controls the difficulty and hence the rate at which new coins are generated. Older block data and a nonce are combined to calculate this hash. Given that cryptographic hash functions are not invertible, in order to find a new valid block the only alternative would be to obtain different nonces until one which fulfills the pre-established requirement is found.

1.2 The Linkability/Traceability Problem

As the authors in [36] pointed out, it is possible to discover the identity of someone who makes a transaction in Bitcoin through traffic analysis and tracing of users IP's. Owing to Bitcoin's design, the first person to publicize a transfer will probably be the payer. Therefore, discovering who the first person to publicize it was, will permit in all probability to know who the payer in the transaction and who the owner of the input addresses used are.

Another type of analysis, which stands out, is that based on the relations which may be established between addresses. Various studies have developed heuristics to reduce the degree of anonymity of Bitcoin users [1, 30]. For example, the authors in [1] estimate that approximately 40 percent of Bitcoin users could be identified using the heuristics set out in the study. A striking result, obtained by applying this type of measure, is that published in [30], in which a relation between the founder of Silk Road and

someone who was probably one of the creators of Bitcoin was established. The tracing capability shown by the researchers remains completely valid.

The above mentioned traceability problems are due to the fact that the input and output addresses of the Bitcoin users are linkable by one way or another. Therefore, protocols to enhance Bitcoin users' anonymity, ensuring unlinkability of input-output addresses, must be provided. Based on what are known as mixing services [7], recently, some efforts have been made towards overcoming the above attacks and providing stronger privacy to the Bitcoin users by mixing multiple transactions to make input and output addresses of transactions unlinkable to each other. In this direction, some third-party Bitcoin mixing services were first to emerge, but they have been prone to thefts.

2 Previous Work

Among the early proposals, to provide unlinkability between individual Bitcoin transactions, without introducing a trusted party, is the ZeroCoin scheme of [31], which is an extension to Bitcoin. It employs a cryptographic accumulator of minted zero-coins and a zero-knowledge proof of inclusion of a certain zero-coin within the accumulator. ZeroCoin suffers from significant computation and communication overheads where, the size of the proof that has to be stored in the blockchain for each transaction is prohibitively large and far exceeds the size of the Bitcoin transaction itself. The scheme is considered incompatible with the standard Bitcoin ecosystem. Later, several contributions were aiming to reduce the ZeroCoin overheads [12, 17, 38]. However, all these schemes require a modification to the Bitcoin standards and hence, they are incompatible with Bitcoin and are unlikely to be accepted for implementation as mentioned in [40].

Anonymous Bitcoin payments are facilitated in the MixCoin scheme [6]. This scheme does not make any modifications to the standard Bitcoin protocol. Central accountable mixing server is employed, where Bitcoin users send their coins. The center in turn, replies with a guarantee of returning the funds to the user. This strategy ensures unlinkability between the user's input and output addresses, since, the mix sends the coins back to the user. However, using this strategy, the decentralized property (the main property of Bitcoin) is eliminated. Unlinkability is only guaranteed against external observers, because the mixing server learns which address belongs to which user.

A modification to the CoinJoin protocol using blind signatures in the work of [29] avoids the problem of a centralized mix learning the relation between input and output addresses. The anonymous communication network, "Tor" [13] is employed in [29] to avoid centralized mixing and to provide unlinkability.

The CoinShuffle protocol in [37], employs the group transactions service in Bitcoin to ensure correctness and thus shares almost the same disadvantages with CoinJoin, i.e., limited anonymity levels and potential transaction fees. CoinShuffle improves over CoinJoin by using decryption mixnets for address shuffling which achieves anonymity against insiders. However, in CoinShuffle the last peer is in the unique position to determine the outcome of the shuffling and might exploit this to select preferred input addresses to her own output addresses. Our scheme – although employs a shuffling subroutine – does not suffer this disadvantage, since all the coins are transfered and committed to an aggregation temporary address before the result of the shuffling is announced.

2.1 Recent Protocols

The most recent protocol (until the time this paper was written) is CoinParty [40], which invokes the threshold ECDSA digital signature protocol of [25] and employs mixing peers to provide unlinkability for the Bitcoin ecosystem. They introduced an elegant idea of committing each input peer to a temporary Bitcoin address where the signature key of this address is jointly shared among the mixing peers on a threshold bases. Each input peer P_i must first transfer the coins x from his input address I_i to a temporary address T_i, controlled by the mixing peers. This strategy commits the input peers to the transactions. Next, after all transactions are confirmed, the mixing peers are responsible for mixing (shuffling) the output addresses and broadcast the shuffled version. Finally, the peers join to sign a transaction from every temporary address T_i to every shuffled output address $O_{\pi(i)}$. The work in this paper is inspired by the CoinParty and the CoinShuffle protocols.

2.2 Existing Vulnerabilities

The work in this paper is motivated by the security vulnerabilities and efficiency drawbacks of recent protocols described next.

CoinParty. Considering CoinParty protocol [40], the employed mixing peers are different from the input peers, therefore, a major drawback is that, once the input peers transfer their coins to the temporary addresses owned by the mixing peers, saboteur mixing peers may runaway (due to halting or disconnection) leaving the coins of the input peers stuck in the temporary addresses with absolutely no way to return them back. The consequences are disastrous if the number of disconnected mixing peers exceeds a certain threshold required to jointly sign a transaction. Moreover, they can easily conspire to steal the coins. In CoinParty, the mixing peers have no financial shares (coins) in the mixing protocol, therefore, they will not loose anything if they attempt such sabotage attack. In other words, they do not care if the input peers loose their coins. Notice that the signing keys of the temporary addresses are shared among the mixing peers on threshold basis. Hence,

if enough number of mixing peers leave, the signing keys are lost. Moreover, they most probably will not tend to participate without incentives (return fees), similar to the Bitcoin miners in a standard Bitcoin ecosystem, which increases the fees paid by the input peers. A serious efficiency drawback in CoinParty is that, the protocol requires the mixing peers to invoke the threshold ECDSA key generation protocol and the threshold ECDSA digital signature protocol for every temporary address. Invoking the threshold ECDSA for every temporary address results in a relatively high computation complexity for each mixing peer, specially, if these protocols are supposed to run on mobile devices. Moreover, the threshold ECDSA they invoked is in the honest-but-curious scenario, giving a great chance for undetected malicious behavior by the peers due to a corruptive adversary. The authors stated that the protocol must be implemented to withstand malicious behavior.

CoinShuffle. CoinShuffle protocol [37] employs the idea of multiparty private shuffle [7, 10, 15, 32]. The problems with the CoinShuffle protocol are as follows: In order to prevent malicious peers from aborting the protocol after they have received their funds thereby leaving another peer unpaid, the protocol performs a single atomic bulk transaction from all input addresses to all shuffled output addresses at once. Although this strategy is better than CoinParty to protect against sabotage attacks, the resulting anonymity set of the mixing is limited to the number of users participating in a particular mixing operation. Also, this bulk transaction makes the peers easily identifiable on the Bitcoin ledger, since it is very rare and almost unused for other purposes. The transaction itself is of large size and hence, dramatically increases the return fees to the Bitcoin miners. Another problem is that, at the end of the shuffling protocol, one peer knows all the output addresses in the clear, including his own, and this knowledge is before the shuffled addresses are known to other peers. This gives a great chance to this peer to reorder the output addresses in a certain way that he can benefit from.

3 Our Contribution

We propose SecureCoin, as a protocol fully compatible with the Bitcoin ecosystem, to realize anonymity and unlinkability security services to Bitcoin peers. Unlike CoinParty, our protocol does not involve any separate mixing peers as helpers and hence, it is possible to avoid sabotage attacks that could be attempted by mixing peers. Only the input peers are the participants in the protocol. Our protocol uses the inherited "aggregation" and "distribution" transactions forms of the Bitcoin and hence, improves complexity of running multiple instants of a threshold digital signature protocol by reducing to only one instant.

SecureCoin protocol improves security and efficiency over the CoinShuffle protocol, such that, it avoides bulk transaction from multiple input addresses to multiple output addresses. Also, it solves the CoinSuffle problem of allowing a particular peer to control mapping of certain input addresses to certain output addresses. The proposed SecureCoin protocol provides protection against sabotage attacks, attempted by any number of participating saboteurs.

The proposed SecureCoin is robust against malicious behavior of minority (one third) of the participating peers. It does not allow any number of participating saboteurs to maliciously behave without either, being detected and revoked, or loosing their input coins. The behavior of the participating peers is indistinguishable from other normal transactions and hence, cannot be distinguished by miners and ledger observers.

We analize the security properties of SecureCoin, evaluate its performance and compare to recently proposed protocols. We show that, SecureCoin is more efficient and requires less fees by the Bitcoin ecosystem.

4 System Model

In this section, we describe the assumptions, model and goals of the protocol presented in this paper.

4.1 Assumptions and Model

We assume the existence of n peers, $P_1, \cdots, P_n$, where $n > 3t$ and $t \geq 1$ is a particular predefined threshold. There is at most t possible malicious peers. We assume a static adversary model, yet, security against adaptive adversary could be achieved by employing a suitable non-committing encryption scheme. Also, security against coercive adversaries could be realized by employing a suitable deniable public key encryption scheme (e.g. [22, 23]).

Each peer has his own public/private key pair. These keys allow the peers to realize authenticated and private channels among themselves. We emphasize that, these keys are different from the ephemeral public/private keys used in the shuffling stage of our protocol. Given a certain threshold t, we assume that there is at least $n = 3t + 1$ peers that survive till the end of the protocol with at most t of them may behave maliciously [11, 25]. We use n to refer to the number of peers remaining after the execution of any phase in our protocol. In our protocol, there are four different classes of public/private key pairs for a public key cryptosystem, each peer holds:

1) The personal public/private key pair to realize private and authenticated channels with other peers.

2) The ephemeral public/private key pair for the oblivious shuffling stage.

3) The partial signature key corresponding to the aggregation address A.

4) The key pairs for his bitcoin pseudonyms (input and destination addresses).

4.2 Problem and Goals

We solve the problem where n peers, $P_1, \cdots, P_n$ (each peer P_i has a certain amount of x coins available at input address I_i) want to transfer that amount from the set of input addresses $I_1, \cdots, I_n$, to a set of destination addresses, $D_1, \cdots, D_n$, such that,

Correctness/safety. Each peer P_i receives back x coins on his destination address D_i. Coins must not be lost, stolen, or double-spent by any peer even in the presence of a malicious adversary. Honest peers should receive their funds in a timely manner.

Anonymity. Input and destination addresses are unlinkable, i.e., only peer P_i can map his own input address I_i to his own destination address D_i.

Protection against saboteurs. Participating peers may be accidentally or deliberately halted (or disconnected) from the network at any stage of the protocol. We argue that, if such sabotage occurs, then all peers must be equally affected. This type of attack may occur easily due to a halting adversary and is easier than corruptive attacks that requires collaboration of peers in harmony in order –for example– to steal other peers coins. In sabotage attacks, we have two situations: The number of saboteurs do not exceed the threshold t. In this case according to secure multiparty computations (SMC) settings, the protocol execution continue normally. The other situation is when the threshold is exceeded and the disconnected peers do not reconnect. This case is really disastrous. As the threshold is exceeded, the SMC protocol will be terminated. Some peers will loose their coins for ever while others may abort without any loss. Our objective in this case is to ensure that, disconnected peers, at any phase of the protocol, will suffer the same amount of loss as the other participating peers. This motivates disconnected peers to rejoin the protocol execution to save their coins. CoinShuffle effectively withstands saboteurs, since it is not based on threshold cryptosystems and the shuffled transfer of funds is made in a one n-to-n bulk transaction. On the contrary, CoinParty is vulnerable to this attack, as stated earlier, due to the incorporation of mixing peers, that have no financial share in the protocol, other than mixing fees as incentives. We conclude that a protocol must be designed such that, any attempt of sabotage attack by any number of participating saboteur peers at any stage of the protocol must cost them the same amount of lose as other participating peers.

Robustness & Revocation. Any malicious behavior attempted by any peer must be detected as early as possible, this malicious peer must be identified and kicked out (revoked) from the protocol, without affecting the good peers progress in the protocol.

Indistinguishability. The transactions performed by the peers incorporated in the anonymous protocol, must not be distinguished from normal Bitcoin transactions, performed by individual peers.

Unforgeability. An adversary must not be able to create a pair consisting of a message (Bitcoin transaction), m and a signature that is valid for m, that has not been generated by a legitimate signer and that passes the verification of the Bitcoin ecosystem.

Fairness. A participating peer that deposit his coins in the aggregation/temporary address must receive his coins, either in his destination address, or returned back to his input address in case he was kicked out of the protocol due to a malicious behavior.

Performance. The protocol should scale to large numbers of peers without imposing prohibitive overheads upon the Bitcoin network.

Compatibility. The mixing protocol must be fully compatible with the current Bitcoin network and produce legitimate Bitcoin transactions.

Cost efficiency. The protocol must be cost-efficient in terms of involved transaction fees.

5 Preliminaries and Basic Tools

In this section, we give an overview of a bunch of basic tools, necessary to build our SecureCoin protocol. These tools are partitioned into two categories: threshold cryptography tools and shuffling tools. The reader needs to be familiar with these tools in order to follow the description of our SecureCoin scheme. First we give an overview on the elliptic curve used in the Bitcoin ecosystem and the standard ECDSA algorithm, then we describe the cryptographic tools employed by SecureCoin.

5.1 Elliptic Curves for the Bitcoin

Standardized elliptic curves that are used most commonly in real-world applications are mostly given in their short Weierstrass form, $E : y^2 = x^3 + ax + b$ and are defined over a finite field $\mathbb{F}_p$, where $p > 3$ is a prime and $a, b \in \mathbb{F}_p$. For non-singular curves, there is a requirement that, $4a^3 + 27b^2$, is non-zero. Given such a curve E, the cryptographic group that is employed in protocols is a large prime-order subgroup of the group $E(\mathbb{F}_p)$ of $\mathbb{F}_p$-rational points on E. The group of rational points consists of all solutions $(x, y) \in \mathbb{F}_p^2$ to the curve equation, together with a point at infinity, denoted by $\mathcal{O}$, the neutral element. The number of $\mathbb{F}_p$-rational points is denoted

by $\#E(\mathbb{F}_p)$ and the prime order of the subgroup by q. A fixed generator of the cyclic subgroup is usually called the base point and denoted by $G \in E(\mathbb{F}_p)$.

For 256-bit primes, in addition to the NIST curve defined over $\mathbb{F}_{p^{256}}$, SEC2 also proposes a curve named secp256k1 defined over $\mathbb{F}_p$ where $p = 2^{256} - 2^{32} - 977$. This curve is the one used in Bitcoin [5].

There are three basic point operations on elliptic curves.

1) Point addition: Let $P_1 \in E(\mathbb{F}_p)$ and $P_2 \in E(\mathbb{F}_p)$, then, $P_3 = P_1 + P_2 \mod p$ is also a point on the elliptic curve, that is, $P_3 \in E(\mathbb{F}_p)$.

2) Point doubling: Let $P \in E(\mathbb{F}_p)$, then $Q = 2P \mod p$ is a point doubling operation where $Q \in E(\mathbb{F}_p)$.

3) Scalar multiplication: Let $k \in Z_q$ and $P \in E(\mathbb{F}_p)$, then, $Q = kP \mod p$ is the process of adding P to itself k times, where $Q \in E(\mathbb{F}_p)$. This process is performed through the well-known double-and-add operations.

5.2 Elliptic Curve Digital Signature

The Elliptic Curve Digital Signature Algorithm (ECDSA) was standardized in FIPS 186-4[1]. The signer generates a key pair (s, Q) consisting of a private signing key $s \in_R Z_q^*$ and a public verification key, $Q = sG \mod p$, where G is the generator point. To sign m, the signer chooses a per-message random integer $k \in_R Z_q^*$, computes the point $(x, y) = kG$, and computes $r = x \mod q$. The signature of a message M, is the pair (r, w), of integers modulo q, where $w = k^{-1}(m + sr) \mod q$ and m is the hash of M.

It is important that the per-message secret k is not revealed, since otherwise the secret signing key s can be computed by $s = r^{-1}(kw - m) \mod q$, because r and w are given in the signature and m can be computed from M. Even if only several consecutive bits of the per-message secrets for a certain number of signatures are known, it is possible to compute the private key (see [26]). Also, if the same value for k is used to sign two different messages M_1 and M_2 using the same signing key s and producing signatures (r, w_1) and (r, w_2), then k can be easily computed as $k = (w_1 - w_2)^{-1}(m_1 - m_2) \mod q$, which then allows recovery of the secret key.

5.3 Threshold Cryptography Tools

In this subsection, we review the threshold cryptography tools that will be used in building our SecureCoin. All theses tools are based on the well-known polynomial/Shamir's secret sharing scheme. We describe the elliptic curve version of these tools, in order to be directly applied for the implementation of SecureCoin.

[1]PUB FIPS. 186–2. digital signature standard (dss). US department of commerce/national institute of standards and technology.

5.3.1 Polynomial Secret Sharing

Consider a secret value, $s \in Z_q$ which is held by a dealer, where Z_q is a prime field. To share this secret among a set $\mathcal{P} = \{P_1, \cdots, P_n\}$ of $n > t$ participants, where t is a certain threshold, the dealer constructs a polynomial $g(x) = \sum_{j=0}^{t} a_j x^j \mod q$, he sets $a_0 = s$ and each other coefficient $a_{j\neq 0} \in_R Z_q$. $\forall i = 1, \cdots, n$, the dealer secretly delivers $g(i)$ to participant P_i. To reconstruct the secret s, each participant $P_i \in \mathcal{P}$ broadcasts $g(i)$, the participants compute s from any $t + 1$ shares using Lagrange interpolation formula, $s = g(0) = \sum_{i\in\mathcal{B}} \lambda_i g(i) \mod q$ where $\mathcal{B} \subset \mathcal{P}$, $|\mathcal{B}| = t + 1$ and, $\lambda_i = \Pi_{j\in\mathcal{B}, j\neq i} \frac{j}{j-i}$, is participant P_i's Lagrange coefficient.

5.3.2 Elliptic Curve Verifiable Secret Sharing

Verifiable secret sharing (VSS) is an extension to polynomial/Shamirs secret sharing to allow the recipients of the secret shares to verify that the shares are consistent (i.e., that any subset of $t + 1$ shares interpolate to the same unique secret). Assuming $n \geq 2t + 1$, the scheme tolerates the malicious behavior of at most t of the n participants. Two different types of VSS are distinguished; the conditionally secure scheme due to Feldman [14] and the unconditionally secure scheme due to Pedersen [34]. For best security, both of them will be used in our SecureCoin protocol. We present an overview of these subroutines over elliptic curves.

EC Feldman-VSS. Let p and q be two large primes, such that $q|p - 1$. The two primes p and q and the base point (EC generator point) G of order q are published as the system public parameters. The dealer shares the secret s among the participants on a t-degree polynomial $g(x) = \sum_{j=0}^{t} a_j x^j \mod q$, the dealer also broadcasts the $t + 1$ commitments $C_j = a_j G \mod p \; \forall j = 0, \cdots, t$. These commitments allow each participant P_i to verify the consistency of his share $g(i)$ by checking that, $g(i)G = \sum_{j=0}^{t} i^j C_j \mod p$. If this check fails for any share $g(i)$, P_i broadcasts a complaint. If more than t participants broadcasted a complaint, then at least one of them is honest, therefore, the dealer is deemed corrupt and disqualified. Otherwise, the dealer broadcasts the share $g(i)$ of each complaining participant P_i, if the share is consistent, P_i is disqualified, otherwise, if the share is inconsistent with the commitments or the dealer does not respond, then the dealer is disqualified. In the reconstruction phase of the secret, all the participants are able to check the validity of the share broadcasted by any of the other participants by verifying with the published commitments to filter out inconsistent shares and safely perform the Lagrange interpolation. When it comes to the distributed generation of a secret key s and the joint computation of $sG \mod p$, Feldman-VSS alone is not secure due to the attacks attempted in [19, 24].

EC Pedersen-VSS. The trick is to perform double exponentiation to allow randomization of the broadcasted commitments. The public parameters in this VSS are p, q and G as in Feldman-VSS and another generator point H, subject to the condition that $\log_G H$ is unknown and assumed hard to compute. In addition to the polynomial $g(x) = \sum_{j=0}^{t} a_j x^j \mod q$ with the secret s as the free term, the dealer constructs another polynomial as a randomization t-degree polynomial $r(x) = \sum_{j=0}^{t} b_j x^j \mod q$. He secretly delivers $(g(i), r(i))$ to participant P_i $\forall i = 1, ..., n$. The dealer also publishes the commitments $C_j = a_j G + b_j H \mod p \; \forall j = 0, \cdots, t$. Each participant P_i is able to verify the consistency of his share $g(i)$ by checking that, $g(i)G + r(i)H = \sum_{j=0}^{t} i^j C_j \mod p$. If this check fails for any share $g(i)$, P_i broadcasts a complaint. If more than t participants broadcast a complaint, then at least one of these participants is honest about his complaint and the dealer is disqualified. Otherwise the dealer broadcasts the pair $(g(i), r(i))$ for each complaining participant P_i, if the pair is consistent, P_i is disqualified, otherwise, if the pair is inconsistent with the commitments or if the dealer does not respond, then the dealer is disqualified. During reconstruction, any participant can verify the validity of the share broadcasted by any other participant via the published commitments to reject invalid shares and correctly computes the interpolation.

5.3.3 Joint Secret Sharing

Joint secret sharing are schemes to allow the participants to jointly share a secret among themselves in the absence of a dealer.

Joint random secret sharing (JR-SS). JR-SS [26] allows a set of n participants to jointly share a random secret among themselves without the assistance of a dealer. Each participant $P_i \in \mathcal{P}$ chooses a random integer $k_i \in Z_q$ and plays the dealer's role to share k_i among the participants over a t-degree polynomial $g_i(x) = k_i + \sum_{j=1}^{t} a_j x^j \mod q$. Each participant $P_i \in \mathcal{P}$ simply sums the shares he receives from the other participants to compute a share $g(i) = \sum_{j=1}^{n} g_j(i)$ which is a point on a t-degree polynomial $g(x)$ with its free term equals a random secret $k = \sum_{i=1}^{n} k_i \mod q$.

Joint random verifiable secret sharing (JR-VSS). To withstand malicious behavior of at most $t < n/2$ participants during the JR-SS, JR-VSS combines JR-SS with Feldman or Pedersen-VSS. In this scheme, each participant $P_i \in \mathcal{P}$ chooses a random secret integer $k_i \in Z_q$ and plays the dealer's role in the VSS protocol to share this secret among the other participants. Complaints are solved as in the VSS scheme. Finally, each participant sums what he has to compute his share on a t-degree polynomial, $g(x)$ with its free term $g(0) = \sum_{i=1}^{n} k_i \mod q$.

Joint zero secret sharing (JZ-SS). This scheme is a special case of the JR-SS. As implied by the name of the scheme, the random secret shared by each participant is a zero. After execution of a JZ-SS scheme, each participant holds a share $g(i)$ on a t-degree polynomial $g(x)$ with its free term $g(0)$ equals zero. This scheme is always employed when we need to randomize the shares, without changing the value of the secret.

Joint zero verifiable secret sharing (JZ-VSS). Similar to JR-VSS, to with stand malicious behavior of at most $t < n/2$ participants in the JZ-SS, the JZ-VSS combines the JZ-SS with Feldman- VSS or Pedersen-VSS. In this scheme, each participant $P_i \in \mathcal{P}$ plays the dealer's role in the VSS protocol to share a zero among the other participants. Complaints are solved as in the JR-VSS protocol. The shares are computed by each participant sums what he has to compute his share on a t-degree polynomial, $g(x)$ with its free term equals zero.

5.3.4 Joint Verifiable Multiplication (JVM) of Shared Secrets

Consider two secret values a and b, respectively shared over t-degree polynomials $A(x)$ and $B(x)$, the joint verifiable multiplication subroutine [4] computes $\mu = ab \mod q$ in a robust and secure way with no information revealed about neither a nor b. Each participant P_i locally computes $C(i) = A(i)B(i) \mod q$ which is a share on a $2t$-degree polynomial $C(x) = A(x)B(x) \mod q$ with $C(0) = \mu$. There is still a security problem; publishing and interpolating the shares $C(1), \cdots, C(n)$ reveals information about $A(x)$ and $B(x)$, therefore, it is necessary to randomize the shares of $C(x)$. To randomize the shares without changing the secret value $C(0)$, the participants run JZ-VSS to share a zero over a $2t$-degree polynomial $R(x)$ with $R(0) = 0$. Each participant P_i finally computes and broadcasts $D(i) = C(i) + R(i)$. The result μ could be computed by interpolating the $2t$-degree polynomial $D(x)$ using the Berlekamp-Welch decoder[2] to filter out bad shares. Since we are interpolating a polynomial of degree $deg = 2t$ and we have a maximum of t malicious participants (i.e.there are at most t possible faults), the Berlekamp-Welch bound implies that, the number of shares needed in order to correctly interpolate the polynomial is at least $deg + 2faults + 1 = 4t + 1$. Consequently, we need $n > 4t$. This bound on n could be reduced to $3t + 1$ if the participants run a polynomial degree reduction subroutine just before applying the Berlekamp-Welch decoding.

[2] L. R. Welch and E. R. Berlekamp. Error correction for algebraic block codes. Google Patents, December 30 1986. US Patent 4,633,470.

5.3.5 Joint Verifiable Reciprocal (JVR) of a Shared Secret

In our SecureCoin, we are faced with the following problem. Given some secret value k, shared among the participants, compute shares of the reciprocal of $k \mod q$ while k is kept secret. Each participant P_i already holds a share $g(i)$ representing a point on a t-degree polynomial $g(x)$ with $g(0) = k$. To compute shares of $k^{-1} \mod q$, we need $n > 4t$ participants to run the reciprocal protocol [3] (or $n > 3t$ in case polynomial degree reduction is employed) as follows: (i) The participants run the JR-VSS, which results in each participant holds a share $v(i)$ of a random secret v over some polynomial of degree t. (ii) The participants run the JVM subroutine and reconstruct $\mu = kv \mod q$, with no information revealed about k or v. (iii) Each participant P_i computes his share of the reciprocal as $\mu^{-1}v(i) \mod q$, which is a share over a t-degree polynomial with its free term equals $k^{-1} \mod q$.

5.4 Mixnets to Multiparty Secret Shuffle

Since the introduction of Chaums mixing network, mixnet [9] and dinning cryptographers problem, DCnet [8] thirty years ago, a number of anonymous authentication protocols have been developed. The mixnet family schemes use a set of mix servers that mix the received messages to make the communication paths ambiguous. The security of mixnet is based on the trust relationship of the mixers, and cannot provide unconditional anonymity.

In [35], inspection of Chaum's scheme [9] showed that the scheme is linkable. In Chaums scheme, the encryption function was assumed to be a one-way trapdoor permutation, such as the textbook version of RSA scheme. As a result, anyone can take an output message, encrypt it again and check with the input messages they obtain. In this way, the mix can be reversed. To prevent this re-encryption and possible size matching of the incoming flow and output flow, all messages are resized through random string padding to be of the same size. The output messages from the mix will be indistinguishable to adversaries, and therefore we can prevent traffic analysis of network transmissions. This will also ensure that no item is processed more than once. By discarding the repeated input messages, replay attacks can also be prevented. Otherwise, an attacker can repeat the input message and observe which output message is repeated. In this way, the relation of input messages and output messages can be discovered and the claimed anonymity is lost.

Several mixnets have also been designed based on zero-knowledge proofs and stronger security assumptions to guarantee delivery or to detect and exclude misbehaving participants. These schemes include flash mixes [27], hybrid mixes [28, 33], and provable shuffles [7, 15, 32]. The Dissent scheme [10] for anonymous messaging allows a group of participants to communicate messages in a private and anonymous way through verifiable secret shuffling.

6 Our SecureCoin Protocol

In this section, we describe in details our SecureCoin Protocol. Our protocol runs in three stages, each consists of few phases:

Stage 1. Aggregated temporary deposit.

- **Phase 1.1.** Distributed generation of A.
- **Phase 1.2.** Joint deposit of the coins.

Stage 2. Shuffling of destination addresses.

- **Phase 2.1.** Destination addresses generation.
- **Phase 2.2.** Oblivious shuffling.
- **Phase 2.3.** Accusations resolution (if exist).

Stage 3. Coins distribution.

6.1 Aggregated Temporary Deposit

The goal of this stage is to allow the peers to aggregate and deposit the required coins in a temporary aggregation address (ECDSA public-key), A, as a form of commitment. The aggregated coins must not be spent by any individual peer that could behave maliciously to steal the aggregated coins. Hence, the private key corresponding to this address A must be protected against minority of malicious peers. To achieve this, the corresponding private key is generated on a verifiable threshold bases and the public-key A is jointly computed such that, none of the peers has any information about the private key. Yet, still a transaction from this address A can be performed through threshold computation of the signature.

6.1.1 Distributed Generation of A

The peers jointly generate the aggregation address, A, as follows:

Step 1. The peers execute JR-VSS with Pedersen-VSS commitments. The execution results in each peer P_i holds a share $S(i)$ of a secret $s \in_R Z_q^*$ over a t-degree polynomial $S(x)$ with $S(0) = s$.

Step 2. The peers that are not disqualified in the JR-VSS in the previous step publish Feldman-VSS commitments to their shared polynomial. I.e., if $S_i(x) = s_i + \sum_{j=1}^{t} a_j x^j$ is the polynomial of peer P_i, P_i publishes $s_i G$ and $a_j G \mod p \ \forall j = 1, \cdots, t$.

Step 3. For any peer P_i who receives at least one valid complaint, the other peers join together to reconstruct his polynomial $S_i(x)$ and the values $s_i G$ and $a_j G \mod p \ \forall j = 1, \cdots, t$ in the clear.

Step 4. Finally, the remaining good peers join to safely compute $A = sG = \sum_{i=1}^{n} s_i G \mod p$.

At this point, each peer P_i holds a share $S(i)$ of an ECDSA private-key s over a polynomial of degree t and have jointly computed the temporary aggregation address $A = sG \mod p$. The disqualified peers in the above subroutine are kicked out and prevented from further participating in the rest of the protocol. In Bitcoin, the peer's address is actually a hashed version of the public key. However, for simplicity and wlog, along the work in this paper, we assume that the peer's address is the ECDSA public key.

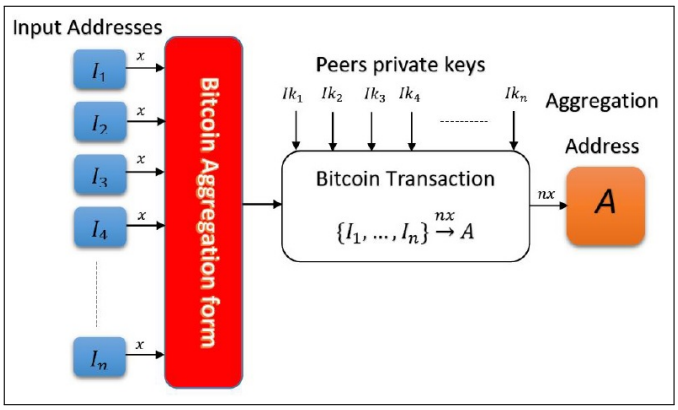


Figure 1: Deposit of the coins in the aggregation address

6.1.2 Joint Deposit of the Coins

The deposit of the coins in the aggregation address A is done in the standard way of Bitcoin transaction. Given that each peer P_i is holding the private key Ik_i corresponding to his input address I_i, the peers jointly generate one single transaction, as shown in Figure 1, containing the peers input addresses, $I_1, \cdots, I_n$, as inputs and the aggregation address A as output, $\{I_1, \cdots, I_n\} \xrightarrow{nx} A$. Notice that a transaction with several input addresses is only valid if it has been signed with all keys belonging to those input addresses [29, 37]. Thus each peer can verify whether the generated joint transaction sends the correct amount of money to the aggregation address, if this is not true, the peer just refuses to sign the transaction. If more than t peers refuse, the protocol aborts without transferring any coins. The peers do not proceed in the execution of the protocol until all the transactions are completed successfully and confirmed on the Bitcoin ledger. For n peers, there must be an amount of nx coins in the address A. The peers that do not contribute their signature are considered opted out of the protocol.

6.2 Shuffling of Destination Addresses

In this stage it is required that every peer P_i generates a fresh ephemeral personal encryption/decryption public key pair (pk_i, sk_i) of an IND-CCA secure public key cryptosystem and broadcasts the resulting public encryption key pk_i.

6.2.1 Destination Addresses Generation

Each peer P_i locally generates for himself a destination address D_i as in the standard Bitcoin address generation, as if P_i is going to make a transaction $I_i \rightarrow D_i$. This address D_i is kept secret for P_i at the moment. Each Participant P_i commits himself to his address D_i by broadcasting $e_i = \varepsilon_{pk_i}(D_i)$ as the public key encryption of his address D_i.

6.2.2 Oblivious Shuffling

The peers shuffle the freshly generated destination addresses, $D_1, \cdots, D_n$, in an oblivious manner [37], similar to the well known mix network of Chaum [30]. This is illustrated in Figure 2. First, the peers are lexicographically ordered according to their input addresses. Each peer P_i uses the encryption keys of each peer $P_{j>i}$ to create a layered encryption of his output address. Then, the peers perform a sequential shuffling, starting with peer P_1: Each peer P_i expects to receive $i-1$ ciphertexts from P_{i-1}. Upon reception, each peer strips one layer of encryption from the ciphertexts, adds her own ciphertext and randomly shuffles the resulting set, according to his picked permutation π. P_i sends the shuffled set of ciphertexts to the next peer P_{i+1}. If everybody acts according to the protocol, the decryption performed by the last peer results in a shuffled list of output addresses. The last peer broadcasts this list.

Each peer checks that his address exists in the broadcasted list and broadcasts a confirmation of existence. If a peer P_i does not find his address in the list he broadcasts a complaint (accusation). The peers enter an accusation resolution subroutine to solve this accusation. Notice that, a peer P_i that does not find his address in the list and keeps silent (i.e., does not broadcast neither a confirmation nor an accusation) is considered halted and is kicked out of the protocol.

6.2.3 Accusations Resolution (If Exist)

The peer P_i who broadcasted a complaint/accusation must be checked for his honesty. P_i is instructed to broadcast his ephemeral private key sk_i. Recall that each P_i has already broadcasted the commitment $e_i = \varepsilon_{pk_i}(D_i)$. The rest of the peers proceed to check the honesty of P_i. Each peer P_j performs as follows:

- Using sk_i, decrypts for D_i.
- Checks whether D_i is in the list of the shuffled destination addresses.
- If D_i is in the list, P_j broadcasts a complaint against P_i.

Finally, in case more than t peers broadcast a complaint against P_i, then P_i is deemed corrupt and kicked out of the protocol. The peers exit the accusation resolution subroutine. Otherwise, the peers declare P_i as honest about his complaint.

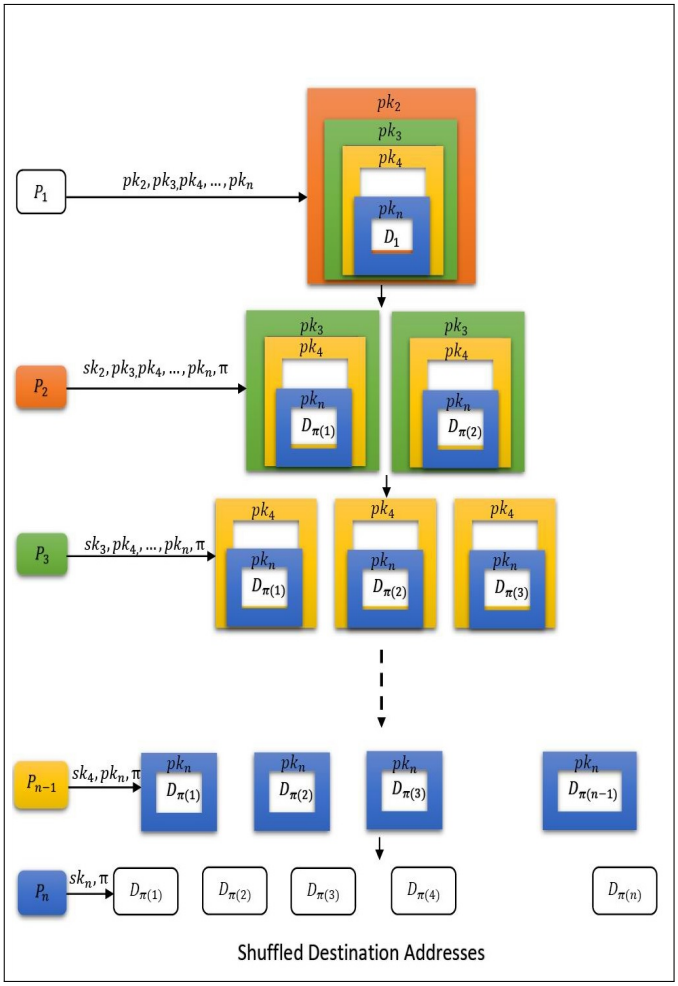


Figure 2: Oblivious shuffling of destination addresses

If the peer P_i passes the above checks, then P_i is declared honest about his complaint, but not necessarily honest about his behavior in the protocol. Still the accusation has not been solved. To proceed in solving the accusation, each peer P_i opens (broadcasts) everything; the destination address D_i, the ephemeral decryption key sk_i and the shuffled patterns. Each peer P_i performs as follows:

- Using sk_j, decrypts for D_j for all $j = 1, \cdots, n$.
- Checks whether D_j is in the list of shuffled destination addresses for all $j = 1, \cdots, n$.
- If the above check fails for any j, P_i broadcasts a complaint against P_j.

If any peer P_j receives complaints from more than t peers, P_j is deemed corrupt and is kicked out of the protocol.

Finally, all peers check the correctness of the performed shuffling and raise complaints against the misbehaved peer. The peers kick out any peer that receives more than t complaints. After all bad peers are kicked out of the protocol, the rest of the good peers repeat the address generation phase, then the shuffling phase, using new fresh pseudonyms and ephemeral keys.

6.3 Coins Distribution

The Coins distribution stage is shown in Figure 3. Let n^* be the number of kicked peers after the deposit of their x coins in the aggregation address. For the n remaining good peers, in this phase, the Bitcoin distribution form (one input to multiple recipients) is used to transfer an amount nx coins from the aggregation address A as a one input address to the output addresses, $D_{\pi(1)}, \cdots, D_{\pi(n)}$, and n^*x coins back to the input addresses, $I_{i_1}, \cdots, I_{i_{n^*}}$ of the peers that were kicked out after their deposit, such that, each address receives an exact amount of x coins. In this case the peers must join to sign the transaction, $T = A \xrightarrow{(n+n^*)x} \{D_{\pi(1)}, \cdots, D_{\pi(n)}, I_{i_1}, \cdots, I_{i_{n^*}}\}$. Let $m = H(T)$ and recall that each peer P_i holds a share $S(i)$ of the ECDSA private key s and that the corresponding public key is $A = sG$, the peers join to sign m and submit as follows.

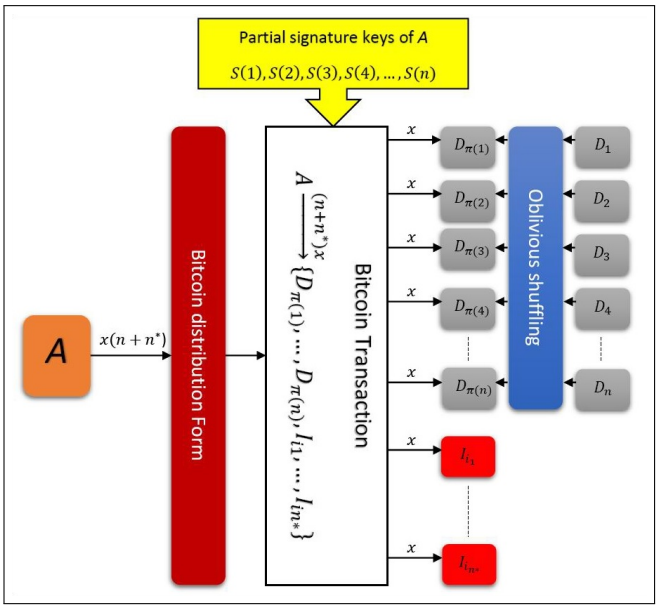


Figure 3: Coins distribution stage

Step 1. The peers execute a Pedersens JR-VSS. At the end, every peer P_i holds a share $K(i)$ of a secret $k \in_R Z_q^*$ over a polynomial $K(x)$ of degree t, with $K(0) = k$. Each peer broadcasts $G_i = K(i)G$ mod p.

Step 2. Each peer broadcasts Feldman-VSS commitments (i.e. to the base G) of all his picked random polynomials during JR-VSS in the previous step. These commitments allow the peers to validate the quantities G_i.

Step 3. Each peer is able to locally compute $(x, y) = kG$ by Lagrange interpolation and $r = x \mod q$.

Step 4. The peers that are not disqualified in the previous JR-VSS run the JVR subroutine, at the end, each peer P_i holds a share $K'(i)$ of the reciprocal k^{-1}

mod q over a t-degree polynomial (with polynomial degree reduction) $K'(x)$ with $K'(0) = k^{-1}$.

Step 5. The peers run an instant of the JZ-VSS to share a zero secret. At the end each peer holds a share $Z(i)$ over a t-degree polynomial $Z(x)$ with $Z(0) = 0$.

Step 6. Each peer P_i now holds a share $S(i)$ of s, a share $K'(i)$ of k^{-1} and a share $Z(i)$ of 0. All peers know the quantities r and m. Each peer P_i locally computes and broadcasts, $\omega_i = K'(i)[m+rS(i)]+Z(i) \mod q$.

Step 7. Using Berlekamp-Welch decoder, any peer is able to compute $\omega = k^{-1}(m + rs) \mod q$.

Step 8. All peers now know the Bitcoin ECDSA signature (r, ω) which is submitted to the Bitcoin system for verification and confirmation in the Bitcoin ledger.

Remark 1. *The above subprotocol requires a number of peers $n > 4t$. This bound on n could be reduced to $n > 3t$, if the peers run a polynomial degree reduction after each run of a JVM subroutine. Simply, assume a secret z shared on a t'-degree polynomial. Each peer shares his share of z over a t-degree polynomial. Finally, each peer sums the shares he receives from the other peers, to obtain a share on a t-degree polynomial for the same secret z.*

7 Security Analysis

In this section we give a rigor analysis of the security of SecureCoin.

7.1 Secrecy, Robustness & Revocation

In the core of the SecureCoin protocol, the joint generation of the private key of the aggregation address A (which is a uniformly distributed random value s) is shared on a threshold basis and the value $A = sG$ is publicly known. The protocol is t-secure, i.e., in the presence of at most t malicious peers:

Correctness.

- Any subset of $t + 1$ valid shares will always reconstruct to the same private key s.
- Any peer is able to locally compute the common public key A.
- The secret s is uniformly distributed in Z_q and hence, A is uniformly distributed in the subgroup generated by G.

Secrecy. A coalition of at most t peers learns no information about s except for what could be implied from the value A itself.

Robustness. From the security of robust threshold cryptography, at most t maliciously active peers will not disrupt the correctness of the protocol and will always be detected and disqualified. Executing the JR-VSS using Feldman-VSS alone has some security vulnerabilities. Malicious peers can deviate the uniform distribution of the result of Feldman's JR-VSS to a non-uniform distribution according to the attack described in [20]. More precisely, in case only Feldman-VSS is used, the attack works as follows: Assume that two traitors peers (say P_1 and P_2) want to bias the distribution towards values of A whose last bit is zero. P_1 gives members, $P_3, \cdots, P_{t+2}$, shares which are inconsistent with his broadcasted values, the rest of the members receive consistent shares. Thus, there will be t complaints against P_1, yet t complaints are not enough for disqualification. The traitors compute $\alpha = \sum_{i=1}^{n} s_i G$ and $\beta = \sum_{i=2}^{n} s_i G$. In case α ends with "0" then P_1 will do nothing and continue the protocol as written. If α ends with "1" then force the disqualification of P_1, this is achieved by asking P_2 to also broadcast a complaint against P_1, which brings the number of complaints to $t+1$. This action sets the public value A to β, which ends with "0" with probability 1/2. An illustrative example is as follows: Let "00", "01", "10", "11" be the possible two MSBs of A. Now, if A ends with zero ("00" or "01") then let α be, but if A ends with one ("10" or "11") then P_2 complains (let β be) and hence the probability that A ends with one (i.e. "11") is 1/2 (notice that "11" is the only case that makes the attack fails), hence, the attack fails with probability 1/4 and so we have. Thus effectively, the traitors have forced strings ending in "0" to appear with probability 3/4 rather than 1/2. One must notice that synchronous broadcast does not prevent such attack to take place. Hence, the third requirement for correctness and the secrecy requirement dramatically fail. Unlike Feldman-VSS, in Pedersen-VSS, the malicious peers view is independent of the value of the secret s, and therefore the secrecy of s is unconditional, which eliminates the possibility of this attack.

Revocation. Our protocol ensures that any malicious peer will be detected as soon as possible and will be revoked from the protocol. In Stage 1, the distributed generation of the aggregation address A employs JR-VSS for sharing the ECDSA private key, which allows every peer to verify the correctness of any quantity he receives from other peers. Any published malicious quantity will be detected by all good peers, if a good peer raises an accusation against a malicious peer, then all good peers will raise this accusation and since there are more good peers than bad peers, the malicious peer (through majority voting) will be revoked. In Phase 1.2, the signing of the transaction with several input addresses is valid if it has been signed with all keys belonging to those input addresses. The signing of each peer is visible to every other peer, therefore, the peer with an

incorrect signature will always be detectable by all other peers and revoked by good peers. In Stage 2, the accusation resolution (Phase 2.3), ensures that any malicious peer during the oblivious shuffling will be detected by other good peers and revoked. The Coins distribution (Stage 3), employs verifiable secret sharing where, again, ensures the detection and revocation of any malicious peer. Step 7 in the Coins distribution employs the Berlekamp-Welch decoder, which is capable of filtering out any faulty shares and ensures the correct computation of the final signature.

7.2 Anonymity and Unlinkability

Unlinkability and randomness depend on the shuffling of destination addresses stage. If there is a raised accusation in the shuffling phase of this stage, the protocol enters the accusations resolution phase where all accusations are solved, malicious peers are kicked out and destination addresses of malicious peers are burnt. Given the threshold t which is the maximum number of possible malicious peers, if at least $t+1$ peers raise an accusation against a certain peer, then at least one peer is honest about his accusation. The employed ephemeral public key encryption is IND-CCA secure, which means that an adversary cannot link destination addresses from the broadcasted ciphertexts $e_1, \cdots, e_n$, as they are randomized by the nature of a CCA secure cryptosystem. Based on observations of the blockchain an attacker can try to guess the mapping between a participant's input and output address. The set of addresses among which the attacker has to guess is the anonymity set and its size is the achieved anonymity level. A larger anonymity set leads to a smaller probability of a correct guess and hence more anonymity. In the following, we analyze peer's anonymity against blockchain observers (outsiders) and against participating peers.

7.2.1 Indistinguishability of Blockchains

We emphasize that, Blockchain observers are outsiders, i.e., they have absolutely no contact with any of the peers participated in the protocol. The SecureCoin protocol uses two types of transaction forms: Aggregation transaction and Distribution transaction. Both of these transaction forms are widely used individually by thousands of Bitcoin peers. Aggregation transactions are used by a peer –for example– to clean his wallet, i.e., when the peer has many pseudonyms in his wallet he uses the aggregation form to transfer his coins simultaneously from different pseudonyms to a single new fresh pseudonym and wipes out the old pseudonyms. Also, they are used when a peer is buying goods which cost a large amount of coins, so he spends the amount of coins from several input addresses of his own to the seller's address. On the other hand, distribution forms are used by an individual peer when he wants to distribute funds (e.g. salary) to multiple employees addresses. SecureCoin uses these two types of transactions in the same standard way of Bitcoin transactions, which make the transactions intended for anonymity by a group of peers in SecureCoin indistinguishable from transactions made by thousands of individual peers over the globe for different purposes.

One problem remains, that may threaten the above indistinguishability. The amount x transfered to/from the aggregation address is fixed. Blockchain observers may observe that (i) some fixed amount x is transfered to an address A' from multiple addresses and (ii) the same amount x leaves this address A' to multiple addresses. Hence, this address A' is more likely to be an aggregation address used for anonymity purpose. Although the observer cannot map an input address to the corresponding destination address, just knowing that these addresses are more likely to be involved in an anonymous transaction (i.e. belong to the same set of peers) is a security vulnerability that must be solved. To fix this problem, notice that it is unlikely that all peers have exactly the same amount x as an unspent transaction. A peer P_i may have some arbitrary x_i as an unspent transaction. In case $x_i < x$, then P_i does not qualify to participate from the very beginning. Now, each peer P_i prepares the transaction $I_i \xrightarrow{x_i} A$ and a return transaction $A \xrightarrow{x_i - x} D_i^*$, where D_i^* is a new fresh pseudonym (destination address) for P_i. The transactions $I_i \xrightarrow{x_i} A$ are signed in the aggregation phase as described while the return transactions $A \xrightarrow{x_i - x} D_i^*$ are signed as part of the transaction in the coins distribution stage to return the change back to their new addresses. In this way, the aggregation and distribution transactions are made with different amount of funds and so the problem is solved.

7.2.2 Indistinguishability of Participating Peers

The participating peers inevitably learn which input and output addresses are involved in the shuffling operation, as they have to sign the corresponding transactions and release them to the Bitcoin network. However, since participating peers do not learn which output belongs to which input address, the anonymity level against participating peers is equal to the remaining number of participants n after all accusations have been solved, which is as good as in the CoinShuffle protocol. Notice that, by nature, the shuffling is insecure if the number of participating peers are less than three. Since, for two peers, one peer will recognize his destination address and immediately maps the other address to the other peer.

7.3 Protection Against Sabotage Attacks

Peers that may withdraw trying – not only to disrupt the progress of the protocol – but to make other participating peers loose their funds with impossibility to return these funds back to their input addresses. It is Ok that the protocol is disrupted and terminated with the coins of each participating peer still in their wallets. In this case, the success of the protocol based on the existence of at

most t malicious peers is accepted. However, it is not accepted at all that the peers may loose their funds given this threshold, while the saboteur peers may run away without loosing the same amount of funds too.

We show that our protocol ensures that such an attempted attack will not succeed without the same loss from the attackers as the good peers. In fact, the sabotage attack becomes serious after the coins has been transfered from the peers accounts to another address. In SecureCoin, once the aggregation address A is established and the malicious peers are disqualified, the remaining peers jointly deposit their coins in A simultaneously. Since individual deposit of each peer allows possible saboteur peers (exceeding the threshold) to withdraw after few transactions have been made, leaving other peers unable to even jointly return their deposits back to their input addresses. Simultaneous deposit ensures that either, the protocol will be terminated without a loss or, the saboteurs will loose their coins too. Therefore, they are unlikely to misbehave after the deposit phase. This is actually a great improvement over the CoinParty protocol, where the mixing peers controlling the temporary address do not have any financial share in the addresses and hence, they may run away leaving the input peers stuck with their coins in the temporary addresses with no way to undo the transactions.

Finally, as has been shown, our protocol does not prevent a sabotage attack to take place, actually it is impossible to prevent it. However, the protocol ensures that if such attack is attempted, then either, no body will loose, or every body will loose.

7.4 Unforgeability

Our protocol employs the ECDSA in the same way used by the Bitcoin ecosystem. To a verifier, the generated signature by our protocol is completely indistinguishable from a signature generated by a Bitcoin wallet. Therefore, given that the Bitcoin signature is unforgeable, and that our threshold key generation is t-secure, our protocol is also unforgeable by an adversary that is able to corrupt at most t peers.

7.5 Deniability

Deniability against outsiders is achieved by the indistinguishability of the SecureCoin transactions and normal transactions. We argue that if there are at any point many more non-SecureCoin aggregated and distributed transactions than SecureCoin transactions in the Bitcoin network, a peer can plausibly deny having participated in SecureCoin. Our inspection of the public ledger shows that there are indeed many non-SecureCoin transactions of the same form as those issued by SecureCoin. Aslo, deniability holds against SecureCoin peers that were kicked out of the protocol prior to the establishment of the aggregation address A. Since, in this case, those kicked out peers do not know neither, the aggregation address nor the input addresses of other peers. However, SecureCoin does not provide deniability against participating peers as long as they knew the aggregation address A. Deniability in this case is an outstanding problem in SMC in general and not due to our protocol. Next we show that the claimed deniabilty against input peers of CoinParty is questionable. CoinParty authors stated that, *Deniability against mixing peers is not achieved because they learn which in-and-output addresses participated in the mixing during the shuffling phase.* Mixing peers also know the identities of the input peers. We argue that mixing peers are outsiders with malicious minority, and hence, the claimed deniability of CoinParty fails.

8 Evaluation and Comparisons

In this section, we evaluate the efficiency of SecureCoin and compare its performance to recent protocols.

8.1 Full Compatibility

The deviation of SecureCoin is transparent to standard Bitcoin clients since, Bitcoin ecosystem is not concerned how the ECDSA keys are generated as long as they are a valid Bitcoin ECDSA key pair. Also, Bitcoin ecosystem is not concerned how the signature is performed as long as the signature is a valid Bitcoin signature and the transaction is in the correct form. In other words, the Bitcoin ecosystem is the verifier while the peer's wallet is the generator of the transaction. Bitcoin has nothing to do with the peer's wallet and how it generates and signs transactions as long as they are in the correct form. In our threshold ECDSA, a verifier receiving a signature cannot distinguish whether this signature is generated by a single signer or by a group of signers on a threshold basis. Hence, SecureCoin is fully compatible with the Bitcoin ecosystem.

8.2 Cost Efficiency

It is known that the processing of a Bitcoin transaction of roughly less than 1KB will not be charged. The amount charged per 1KB defaults to 0.0001 XBT. Let n_i and n_o be the number of input addresses and output addresses of a transaction respectively. The size S of a transaction, assuming compressed public keys, could be roughly estimated based on the simple formula, $S = 148n_i + 34n_o + 10$ Bytes[3]. SecureCoin requires two transactions: An aggregation transaction with $n_i = n$ and $n_o = 1$ and a distribution transaction with $n_i = 1$ and $n_o = n$. For $n = 6$ participants, based on this formula, the first transaction is of size $S = 932$ Bytes while the second is of size $S = 362$ Bytes. Hence, no processing fees are due.

[3] http://bitcoinfees.com/

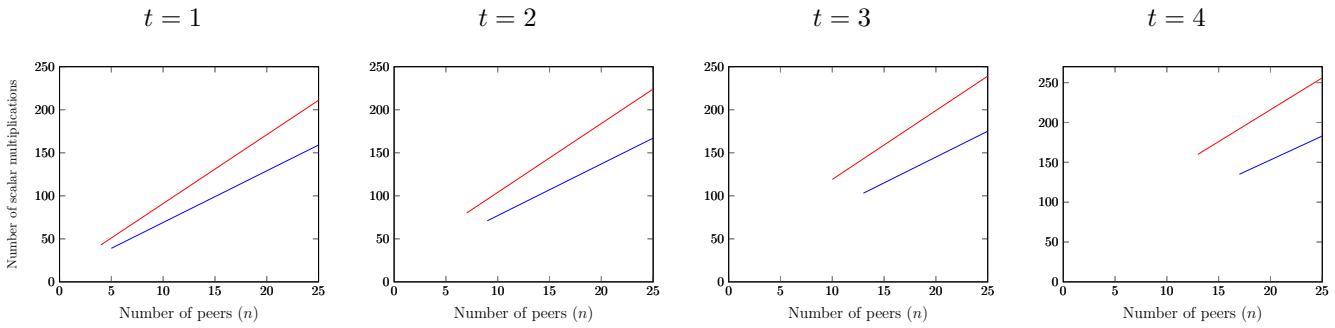


Figure 4: Number of scalar multiplications of SecureCoin for the threshold-ECDSA at different values of the threshold and the number of peers: blue ($n > 4t$), red ($n > 3t$).

8.3 Computation Complexity

Consider the joint generation of the address A. This requires one invocation of a JR-VSS which by its turn requires $2(t+1)+2(n-1)$ EC point multiplications.

Now Consider the threshold ECDSA signature in the Coins distribution stage. In step 1, the JR-VSS with Pedersen-VSS requires $2(t+1)+2(n-1)$ EC point multiplications. Step 2, requires no extra computations by each peer. Step 3, requires each peer to perform Lagrange interpolation on the broadcasted values in step 2, and hence, requires $t+1$ EC point multiplications. Step 4, requires the JR-VSS inside the JVR subroutine over $2t$-degree polynomial, which requires $2(2t+1)+2(n-1)$. In Step 5, the JZ-VSS is similar to the JR-VSS in step 1. Step 6, 7 and 8, requires no EC point multiplications. These total $8t+6n+1$ EC point multiplications that must be performed by each peer. The above computations assume $n > 4t$. In case $n > 3t$ the computation complexity increases due to the employment of polynomial degree reduction in step 4. In this case, it requires a total of $t^2+10t+8n$ EC point multiplications. These are illustrated graphically in Figure 4 for different values of the threshold and number of participating peers.

We notify the recent work of [18] that may provide a slightly improved efficiency for the implementation of threshold ECDSA. However, the protocol in [18], although it is unforgeable, it sacrifices robustness for the sake of efficiency and is not suitable for the goals of SecureCoin.

8.4 Comparisons

In this subsection, we compare the complexity and security of our protocol with previous protocols.

8.4.1 With CoinParty

Based on the transaction size, CoinParty is more efficient since all transactions are one-to-one. However, the employed mixing peers will not provide their services for free. In addition, they are many (at least 4). We cannot give any estimate of the cost because this has not been standardized yet. However, the cost will be relatively high to avoid sabotage.

CoinParty has the major problem of its vulnerability to sabotage attacks. Saboteur peers involved in the mixing and holding the private keys of the temporary addresses may abort the protocol after the input peers make their deposit, leaving them stuck with the impossibility to refund. The saboteurs did not loose any thing, since they have no share in the funds. They also may conspire to steal the coins. Our protocol ensures that if such sabotage attack is attempted, then either, the protocol terminates with all peers, including the saboteurs loosing their funds, or the protocol terminates with no loss at all. This is ensured by phase 1.2 of stage 1, joint deposit ensures that all peers will contribute in the deposit of the same amount x. A transaction with several input addresses is only valid if it has been signed with all keys belonging to those input addresses. Therefore, this phase ensures the simultaneous contribution of funds.

CoinParty runs multiple instants of the threshold key generation and the threshold digital signature (n instants). These protocols are complex by nature as we have shown, so running multiple instants is significantly complex specially if these protocols are supposed to run on smart devices. Figure 5, shows a comparison between our SecureCoin protocol and CoinParty for different values of the threshold and number of peers. It illustrates the dramatic increase in the number of EC point multiplications required in CoinParty over that required by SecureCoin. SecureCoin invokes the threshold protocols only once. This provides a significant complexity improvement over CoinParty.

8.4.2 With CoinShuffle

The cost-efficiency of our protocol proves efficiency over CoinShuffle. This is illustrated in Figure 6. It shows that, for $n = 6$, our protocol requires one aggregation transaction of size 932 Bytes and one distribution transaction of size 362 Bytes. On the other hand the CoinShuffle requires one multi-input multi-output transaction of size 1102 Bytes which is charged. Increasing n, Figure 6 shows that as long as $n < 14$ our protocol charges the participants with only 1KB, while this is limited to $n < 11$ in the CoinShuffle protocol. The CoinShuffle protocol has

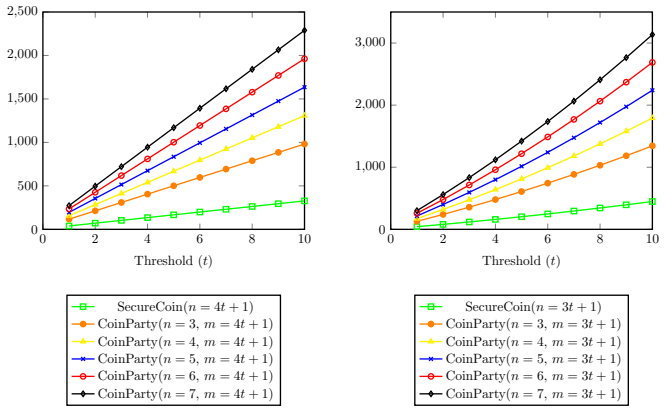


Figure 5: The number of scalar multiplications required by SecureCoin and CoinParty (m=number of mixing peers, n=number of input peers)

a security vulnerability that the last peer in the Shuffle protocol knows the set of shuffled destination addresses in the clear and the set of input addresses. This allows him to rearrange the shuffled version in a way to map certain input addresses to certain output addresses and benefit from this behavior specially if he collaborates with others in the protocol. SecureCoin eliminates this vulnerability since the set of input addresses and the set of destination addresses are isolated by the aggregation address A.

The bulk (exactly n-input to exactly n-output) transaction performed by CoinShuffle for a significant number of addresses is not a commonly used form by Bitcoin and makes it distinguishable by the ledger's observers as this type of transaction is rarely to be performed by an individual. Actually, there is no reason for an individual to perform such a costly transaction. Our protocol avoids this type of transaction. Instead, the aggregation transaction and the distribution transaction used in our protocol are performed frequently by many individuals in the Bitcoin network.

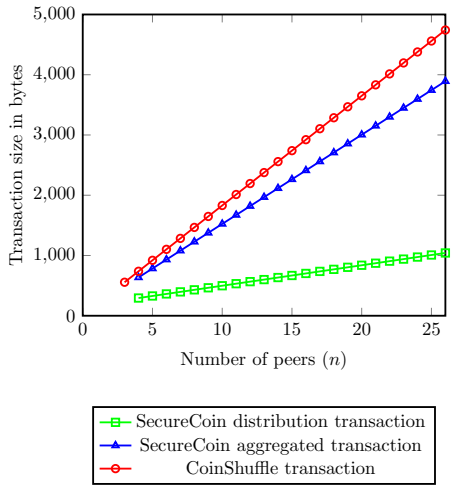


Figure 6: Comparison of transaction size between CoinShuffle and SecureCoin

8.5 Computation Time

Let MM denotes a modular multiplication operation while MA denotes a modular addition operation. The EC point addition (PA) requires 8MM+3MA operations. On the other hand, a point doubling requires 3MM+4MA operations [21]. The most basic technique for performing an EC scalar multiplication (SM), kG for an integer k, is the double-and-add method, which works in a similar way as the square-and-multiply method for exponentiation. Given a scalar k of a length of n bits, the double-and-add approach executes n point doubling and on the average of $n/2$ point additions; the exact number of point additions depends on the Hamming weight of k. Therefore, the overall cost of the double-and-add method to perform SM amounts to $3n + 8n/2 = 7n$ multiplications and $4n + 3n = 2 = 5.5n$ squarings over $\mathbb{F}_p$.

A better strategy for computing kG is to decompose the n-bit scalar k into two half-length integers k_1 and k_2 (often referred to as balanced length-two representation of k [21]). As a result, the overall cost amounts to $(0.5n)(3) + (0.375n)(8) = 4.3n$ multiplications and $(0.5n)(4) + (0.375n)(3) = 3.125n$ squarings in $\mathbb{F}_p$. However, the Hamming density can be reduced to 0.5 (on average) by representing k_1 and k_2 in Joint Sparse Form (JSF) [39], which, in turn, cuts the number of point additions by roughly one third to $0.5(n = 2) = 0.25n$. In this case, the total cost of computing kP is reduced to – on the average of – $(0.5n)(3) + (0.25n)(8) = 3.5n$ multiplications and $(0.5n)(4) + (0.25n)(3) = 2.75n$ squarings in $\mathbb{F}_p$.

To evaluate the computation time of our protocol, an implementation of the basic field arithmetic over prime field $\mathcal{F}_p$ for a 256-bit p secp256k1 on a mobile phone is shown in Table 1. This below-moderate specifications mobile device runs Android OS V2.3 on a CPU 1 GHz Scorpion and 768 MB RAM.

Table 1: Computation time of basic field arithmetic operations on a mobile phone (HTC-Desire)

Operation	Computation time
Multiplication	990.2 ns
Addition	121.2 ns
Subtraction	129.5 ns
Inverse	190.1 μs
Squaring	859.5 ns

Based on Table 1, we can determine the time taken by this device to perform EC operations. The SM operation requires, $(3.5)(990.2$ ns$)(256$ bits$) = 0.88$ ms, plus $(2.75)(859.5$ns$)(256$ bits$) = 0.6$ ms. Hence, a scalar multiplication takes about 1.5 ms. Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ and $P_3 = (x_3, y_3)$ be three points on the elliptic curve. Now let $P_3 = P_1 + P_2$. P_3 is computed as follows: compute $\lambda = (y_2 - y_1) = (x_2 - x_1)$, $v = (y_1x_2 - y_2x_1) = (x_2 - x_1), x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$. All computations are modulo p. Now we can concretely find what it takes to perform one

point addition on the EC. Computing λ requires two modular subtractions and one modular inversion. Computing v (given λ was computed) requires two modular multiplications and one modular subtraction. Computing x_3 requires one modular squaring and one modular subtraction while y_3 requires two modular subtractions and one modular multiplication. These totals six modular subtractions, one modular inversion, one modular squaring and three modular multiplications. From Table 1, we have for one point addition, $6(0.1295\mu s) + 190.1\mu s + 0.8595\mu s + 3(0.9902\mu s) = 0.195$ ms. These are summarized in Table 2.

The above implementation shows that scalar multiplication on elliptic curve is expensive compared to other modular operations. Based on Table 2, Figure 7, illustrates the threshold-ECDSA execution time required by SecureCoin compared to that required by CoinParty. Figure 7, shows that, for a threshold t up to the value of 10, which is a large value, the computations required by SecureCoin is less than one second.

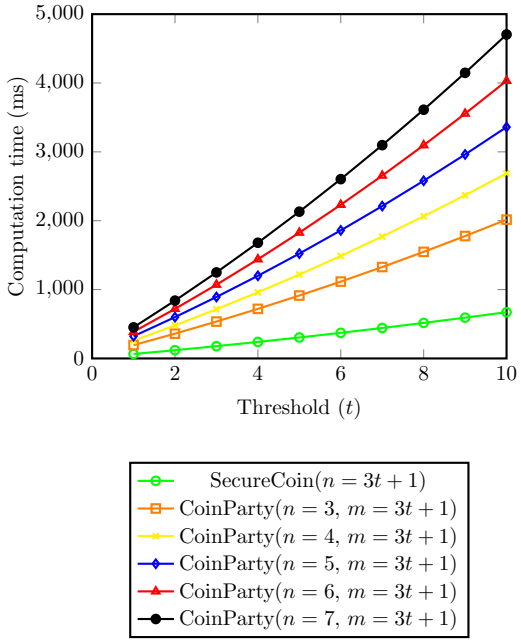


Figure 7: Comparison of Computation time of threshold-ECDSA between CoinParty and SecureCoin

Table 2: EC Computation time on a mobile phone. SM: Scalar Multiplication, PA: Point Addition

Operation	Computation time
SM	1.5 ms
PA	0.195 mss

9 Conclusions

Several contributions have been proposed recently to countermeasure the attacked anonymity of Bitcoin addresses. However, by analyzing these protocols, serious vulnerabilities have been revealed. CoinShuffle performs a bulk transaction of exactly n input addresses to n output addresses of the same amount which is easily observed on the blockchain. In CoinParty, the input peers are exposed to sabotage attacks by mixing peers, and in order to reduce the risk of such an attack, the return fees for the mixing peers are dramatically increased. In this paper, we proposed SecureCoin as a robust and secure protocol for achieving anonymity service in Bitcoin. Our protocol provides better protection for the participating peers against malicious behavior of minority of the peers and protection against the most serious sabotage attack attempted by any number of saboteur peers. We analyzed the security of our scheme and evaluated its efficiency. Finally, we compared our protocol to recently proposed protocols and showed that our protocol proves efficiency over these protocols and requires less fees by the Bitcoin ecosystem.

References

[1] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *Financial Cryptography and Data Security*, pp. 34–51, Springer, 2013.

[2] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, O'Reilly Media, Inc., 2014.

[3] J. Bar-Ilan and D. Beaver, "Non-cryptographic fault-tolerant computing in constant number of rounds of interaction," in *Proceedings of the Eighth Annual ACM Symposium on Principles of Distributed Computing*, pp. 201–209, 1989.

[4] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for noncryptographic fault-tolerant distributed computation," in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pp. 1–10, 1988.

[5] S. Blake-Wilson and M. Qu, *Standards for Efficient Cryptography (SEC) 2: Recommended Elliptic Curve Domain Parameters*, Certicom Research, Oct 1999.

[6] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *Financial Cryptography and Data Security*, pp. 486–504, Springer, 2014.

[7] J. Camenisch and A. Mityagin, "Mix-network with stronger security," in *Privacy Enhancing Technologies*, pp. 128–146, Springer, 2006.

[8] D. Chaum, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *Journal of Cryptology*, vol. 1, no. 1, pp. 65–75, 1988.

[9] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[10] H. Corrigan-Gibbs and B. Ford, "Dissent: accountable anonymous group messaging," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pp. 340–350, 2010.

[11] I. Damgard, M. Geisler, M. Kroigaard, and J. B. Nielsen, "Asynchronous multiparty computation: Theory and implementation," in *Public Key Cryptography (PKC'09)*, pp. 160–179, Springer, 2009.

[12] G. Danezis, C. Fournet, M. Kohlweiss, and B. Parno, "Pinocchio coin: Building zerocoin from a succinct pairing-based proof system," in *Proceedings of the First ACM workshop on Language Support for Privacy-enhancing Technologies*, pp. 27–30, 2013.

[13] R. Dingledine, N. Mathewson, and P. Syverson, *Tor: The Second-generation Onion Router*, Technical report, DTIC Document, 2004.

[14] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *28th IEEE Annual Symposium on Foundations of Computer Science*, pp. 427–438, 1987.

[15] J. Furukawa and K. Sako, "An efficient scheme for proving a shuffle," in *Advances in Cryptology (CRYPTO'01)*, pp. 368–387, Springer, 2001.

[16] R. P. Gallant, R. J. Lambert, and S. A. Vanstone, "Faster point multiplication on elliptic curves with efficient endomorphisms," in *Advances in Cryptology (CRYPTO'01)*, pp. 190–200, Springer, 2001.

[17] C. Garman, M. Green, I. Miers, and A. D. Rubin, "Rational zero: Economic security for zerocoin with everlasting anonymity," in *Financial Cryptography and Data Security*, pp. 140–155, Springer, 2014.

[18] R. Gennaro, S. Goldfeder, A. Narayanan, "Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security,", *IACR Cryptology ePrint Archive*, vol. 2016, pp. 13, 2016.

[19] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," *Journal of Cryptology*, vol. 20, no. 1, pp. 51–83, 2007.

[20] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," *Journal of Cryptology*, vol. 20, no. 1, pp. 51–83, 2007.

[21] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer Science & Business Media, 2006.

[22] M. H. Ibrahim, "A method for obtaining deniable public-key encryption," *International Journal of Network Security*, vol. 8, no. 1, pp. 1–9, 2009.

[23] M. H. Ibrahim, "Receiver-deniable public-key encryption," *International Journal of Network Security*, vol. 8, no. 2, pp. 159–165, 2009.

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

[25] M. H. Ibrahim, I. A. Ali, I. I. Ibrahim, and A.H. El-sawy, "A robust threshold elliptic curve digital signature providing a new verifiable secret sharing scheme," in *IEEE 46th Midwest Symposium on Circuits and Systems*, vol. 1, pp. 276–280, 2003.

[26] I. Ingemarsson and G. J. Simmons, "A protocol to set up shared secret schemes without the assistance of a mutually trusted party," in *Advances in Cryptology (EUROCRYPT'90)*, pp. 266–282, Springer, 1991.

[27] M. Jakobsson, "Flash mixing," in *Proceedings of the Eighteenth Annual ACM Symposium on Principles of Distributed Computing*, pp. 83–89, 1999.

[28] M. Jakobsson and A. Juels, "An optimally robust hybrid mix network," in *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing*, pp. 284–292, 2001.

[29] G. Maxwell, *Coinjoin: Bitcoin Privacy for the Real World*, Bitcoin Forum, aug. 2013. (`https://bitcointalk.org/index.php`)

[30] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 ACM Conference on Internet Measurement Conference*, pp. 127–140, 2013.

[31] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *IEEE Symposium on Security and Privacy (SP'13)*, pp. 397–411, 2013.

[32] C. A. Neff, "A verifiable secret shuffle and its application to e-voting," in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pp. 116–125, 2001.

[33] M. Ohkubo and M. Abe, "A length-invariant hybrid mix," in *Advances in Cryptology (ASIACRYPT'00)*, pp. 178–191, Springer, 2000.

[34] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology (CRYPTO'91)*, pp. 129–140, Springer, 1992.

[35] B. Pfitzmann and A. Pfitzmann, "How to break the direct rsa-implementation of mixes," in *Advances in Cryptology (EUROCRYPT'89)*, pp. 373–381, Springer, 1990.

[36] F. Reid and M. Harrigan, *An Analysis of Anonymity in the Bitcoin System*, Springer, 2013.

[37] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: Practical decentralized coin mixing for bitcoin," in *Computer Security (ESORICS'14)*, pp. 345–364, Springer, 2014.

[38] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. V. Zerocash, "Decentralized anonymous payments from bitcoin," in *IEEE Symposium on Security and Privacy (SP'14)*, pp. 459–474, 2014.

[39] J. A Solinas, *Low-weight Binary Representations for Pairs of Integers*, Technical Report, 2001. (`http://cacr.uwaterloo.ca/techreports/2001/corr2001-41.ps`)

[40] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and K. Wehrle, "Coinparty: Secure multi-party mixing of bitcoins," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, pp. 75–86, 2015.

Maged Hamada Ibrahim received B.Sc. in Communications and Computers Engineering from Helwan University, Cairo, Egypt, with Distinction and Honor's Degree in 1995. He also obtained his M.Sc. from the same University in 2001. Then his Ph.D. from Helwan University in 2005. He is now an Associate Professor at Helwan University. He is joining several network security projects in Egypt. His main interest is engineering cryptography and communications security. More specifically, working on the design of efficient and secure cryptographic algorithms and protocols, in particular, secure distributed multiparty computations, public key infrastructures, digital signatures, digital rights management protocols and non-cryptographic solutions to telecommunication security problems. Other things that interest him are number theory and the inspection of mathematics for designing secure and efficient cryptographic schemes.

Outsource the Ciphertext Decryption of Inner Product Predicate Encryption Scheme Based on Prime Order Bilinear Map

Xingbing Fu[1], Xunyun Nie[1], and Fagen Li[2]

(Corresponding author: Xingbing Fu)

School of Information and Software Engineering, University of Electronic Science and Technology of China[1]
No. 4, North Jianshe Road, Chenghua District, Chengdu, Sichuan 610054, P.R. China
School of Computer Science and Engineering, University of Electronic Science and Technology of China[2]
No.2006, Xiyuan Avenue, West Hi-tech Zone, Chengdu 611731, P.R. China
(Email: fuxbuestc@126.com)

(Received Jan. 20, 2016; revised and accepted Mar 11 & Apr. 17, 2016)

Abstract

In the private index predicate encryption scheme, the ciphertext not only hides the message, but also hides the attributes. Predicate encryption scheme can enforce the fine grained access control over the encrypted data and perform selective search on the encrypted data. However, the main efficiency drawback of predicate encryption scheme is that the size of the ciphertext and the time required to decrypt it scale with the complexity of the predicate. In this work, we proposed a novel inner product predicate encryption scheme with verifiable outsourced ciphertext decryption based on prime order bilinear group, which significantly reduces the overhead of the data consumer. In the proposed scheme, the data consumer provides the cloud server with a transformation key with which the cloud server transforms the ciphertext associated with the attributes which satisfy the predicate associated with the private key into a simple and short ciphertext, and thus it significantly reduces the time for the data consumer to decrypt the ciphertext, whereas the cloud server does not know the underlying plaintext message for any data consumer; simultaneously, the data consumer can check whether the transformation done by the cloud server is correct to verify the correctness of the transformation.

Keywords: Decryption outsourcing, inner product predicate encryption, prime order bilinear map, RCCA security

1 Introduction

Predicate encryption which can enforce fine grained access control over the encrypted data and perform the selective search on the encrypted data is a novel public key encryption paradigm. In predicate encryption scheme, the private key corresponds to the predicate, and the ciphertext is associated with the attribute set. The private key $PriKey_f$ corresponding to the predicate f can decrypt the ciphertext associated with the attribute A, if and only if $f(A) = 1$.

The traditional public key encryption scheme is coarse-grained: the sender encrypt the message m using the public key PK, only if the owner of private key associated with the public key PK can decrypt the ciphertext to recover the plaintext message m. This scheme is suitable for point-to-point communication, and the encrypted data are sent to the recipient whom the sender is known to in advance. Recently, with the advent of cloud computing, the data owner may want to store their sensitive data in the cloud server such that the sensitive data are accessible to the data consumers anytime and anywhere. However, the cloud servers are honest and curious: on one hand, it performs the various services for data owners and data consumers according to the protocol requirements; on the other hand, it may sell the sensitive data belonging to the data owner to his competitors to obtain the economic benefits. Furthermore, the adversary may want to obtain the sensitive data to do damage to the data owner. Therefore, the data owner encrypts the data to preserve the privacy of data, then he outsources the encrypted data to the cloud server to store such that the authorized consumers can access them. If the traditional encryption scheme is used, the data consumer is not able to search the encrypted data stored in the cloud server. In fact, the data consumer needs to download the encrypted data first, decrypts them, and searches them. When the big complex data are processed in the cloud computing environment, this method will bring about the huge processing overhead and communication overhead.

Boneh et al. [2] first investigated this problem, and they

first introduced the encryption scheme that supports the equality tests. In this scheme, the owner of the public key can calculate the trapdoor information K_m for any message m, and the K_m allows for the server storing the data to test whether a given ciphertext encrypts message m on the condition that any additional information is not obtained. They used this scheme to encrypt the e-mails which were stored in the server such that the data consumer only downloaded the e-mail messages with a given subject without downloading the whole messages and decrypting them. Goyal et al. [11] proposed the key policy attribute based encryption scheme. In this scheme, the ciphertext is associated with the attribute sets, and the private key is associated with the predicate. The private key can decrypt the ciphertext if and only if the attribute set associated with the ciphertext satisfies the predicate associated with the key. Their scheme employs secret sharing scheme, hence, their scheme is expressive. Due to expressiveness of attribute based encryption schemes, they are suitable for many cloud computing and cloud storage applications. Lee, Chung, and Hwang [15] surveyed attribute based encryption scheme of access control in cloud environments. Chung, Liu, and Hwang [7] surveyed attribute-based proxy re-encryption scheme in cloud environments. However, these schemes only achieve payload hiding, that is to say, in these schemes, only the plaintext privacy is guaranteed, whereas the attribute set associated with the ciphertext is public, and not hidden, i.e., the privacy of attribute is not guaranteed. In some highly sensitive environments, not only the privacy of message is guaranteed, but also the privacy of attribute is guaranteed. That is to say, the attribute hiding should be guaranteed. For example, in the personal health record, oncologist attribute suggests that someone or person associated with him has tumour. To meet this requirement, Boneh and Waters [3] proposed hidden vector encryption scheme. Their scheme supports conjunctions, subsets and range query. However, their scheme does not support delegation. Shi and Waters [18] proposed hierarchical hidden vector encryption scheme that supports delegation. However, both schemes do not support disjunction query. In order to support disjunction query, polynomial equation, and inner product calculation, Katz et al. proposed inner product predicate encryption scheme, KSW scheme [14]. However, these three schemes are based on composite order bilinear group.Since group operations and especially bilinear map are prohibitively slow on composite order elliptic curves: a Tate pairing on a composite order elliptic curve whose group order is 1024 bits is roughly 50 times slower than the same pairing on a comparable prime order elliptic curve [17]. With the security levels increasing, this performance gap will become worse. To obtain the same security level, in contrast with composite order elliptic curve group, prime order elliptic curve group requires less order. In order to solve the inefficiency on composite order elliptic curve group, Iovino and Persiano [13] proposed hidden vector encryption scheme based on prime order bilinear group. However, their scheme only supports conjunction calculation, and does not support disjunction calculation. In order to improve efficiency, Freeman [8] proposed inner product predicate encryption scheme, and their scheme obtains the same functionality as KSW scheme [14], whereas their scheme is implemented on prime order elliptic curve group. Both KSW scheme [14] and Freeman scheme [8] are only chosen plaintext secure against attack (**CPA**), and are not chosen ciphertext secure against attack (**CCA**), even not replayable chosen ciphertext secure against attack (**RCCA**). In this work, we employ Freeman scheme [8] as a building block, and attempt to improve security and efficiency.

However, the current predicate encryption schemes, whether they are based on composite order bilinear group or on prime order bilinear group, have the common drawbacks: the ciphertext size and time to decrypt it scale with the size of predicate. When the data consumer employs the resource-restrained device to manage and query the private data stored on his device, the increasing requirement is to outsource calculation to the cloud server where you pay as you use.

How can securely outsource the decryption of ciphertext? A naive method is that the data consumer sends his private key $PriKey$ to the cloud server, the cloud server decrypts the ciphertext which is requested by the data consumer, and then it sends the data decrypted to the data consumer, which requires that outsourcing service is fully trusted. In fact, the cloud server provider can employ the data consumer's private key $PriKey$ to decrypt the ciphertext which will be sent to the other data consumers to recover the plaintext message to obtain the economic benefits. Furthermore, the data decrypted are transmitted in the clear; once the attacker captures the data, the confidentiality of data is compromised. The second method is the outsourcing techniques [6, 9] based on fully homomorphism encryption [10]. These schemes outsource the computation to the cloud server, such that not only the input privacy is guaranteed, but also the decryption keys and messages privacy are guaranteed. However, fully homomorphism encryption schemes and those schemes based on fully homomorphism encryption schemes are not suitable for outsourcing due to inefficiency. If secure pairing outsourcing techniques are employed, then pairing calculations are outsourced to servers. However, the scheme [5] requires the client to calculate multiple exponentiations in the target group where every outsourced pairing is performed. These exponentiations are too expensive and the overhead of the client scales with the predicate size. Furthermore, every pairing operation in the original scheme has four pairings which will be done by the proxy such that the client's bandwidth requirements increase as well. Given the aforementioned drawbacks, our scheme outsources the decryption of inner product predicate encryption ciphertext to the cloud server to perform, and imposes the minimal overhead on the data consumer.

In this work, we proposed a novel scheme that securely

and efficiently outsources the decryption of the inner product predicate encryption ciphertext. The proposed scheme significantly reduces the overhead of the data consumers. In this scheme, the data consumer provides the cloud server with a transformation key such that the cloud server can transform the inner product predicate encryption ciphertext into a simple and short ciphertext without the cloud server knowledging the data consumer's plaintext data, and simultaneously the transformation done by the cloud server can be verified to guarantee that the transformation done by the cloud server is correct. The proposed scheme significantly saves the client bandwidth and the local calculation time: the size of the ciphertext transformed is much smaller than the size of the original ciphertext, and the time to decrypt the transformed ciphertext is much less than the time to decrypt the original ciphertext. Therefore, the resource-restrained device consumes less power. Our scheme is secure against the malicious cloud server as well. Furthermore, our scheme achieves **RCCA** security.

The remainders of our paper are organized as follows: We discuss related work in Section 2. We introduce preliminaries in Section 3. We present the syntax, and security model of the proposed scheme in Section 4. We present the architecture of the proposed scheme in Section 5. We present the scheme construction in Section 6. We give the security proof of the proposed scheme in Section 7. The performance of the proposed scheme is evaluated in Section 8. We draw the conclusion in Section 9.

2 Related Work

In this section, we give the related work as follows: interactive verifiable calculation, bilinear pairing delegation, and proxy re-encryption.

Interactive Verifiable Computation. Interactive verifiable computation [6, 9] enables the resource-restrained devices with weak computation to outsource the computation on the functions to a server, the server returns the result of computations to the client, and gives to it the non-interactive proof that the computation on the functions is correct. Since these schemes [6, 9] outsource the computation to the cloud server, and protect the privacy of input data. However, these schemes are based on fully homomorphism encryption. The overheads of these schemes are so large that they cannot be applied to the cloud computing system. Parno et al. [16] proposed the verifiable computation from attribute based encryption. Their scheme obtains the public delegation and public verification. They proposed the multi-function verifiable computation scheme as well, and this scheme is based on attribute based encryption scheme with outsourced decryption ciphertext due to Green et al. [12]. However, these schemes are focused on the delegation of general functions, not on the efficiency of the problems. Furthermore, these schemes are based on attribute based encryption which only achieve payload hiding, that is to say, the privacy of message is guaranteed, whereas the attributes are public. In some highly sensitive settings, the attributes are required to be hidden. Predicate encryption schemes obtain the attribute hiding.

Pairing Outsourcing. Chevallier-Mames et al. [5] proposed the pairing outsourcing which enables a client to outsource the pairing computation to another entity. However, this scheme still requires the client to calculate multiple exponentiations in the target group where every outsourced pairing is performed. If their scheme is employed to outsource the decryption of predicate encryption ciphertext, the overhead of the client will be proportional to the size of the predicate.

Proxy Re-Encryption. The proposed scheme shows that the client allows the cloud server to transform the predicate encryption ciphertext on m into a simple and short ciphertext, whereas the cloud server acting as the proxy does not learn the underlying plaintext message m. This method is similar to proxy re-encryption [1], in which a semi-trusted proxy is given a proxy key which allows it to transform a ciphertext under one public key into a ciphertext of the same message under another public key without learning the underlying plaintext message. However, in the traditional proxy re-encryption scheme, the correctness of the transformation done by the proxy is not guaranteed. Since the proxy can replace the encryption of m under the delegator's public key with the encryption of another message m' under the delegator's public key, and then employs the proxy keys to transform the latter into an encryption of m' under the delegatee's public key, which brings about reducing the significant computation to perform the other computation services to obtain the economic advantages.

3 Preliminaries

3.1 Bilinear Group Generator

A bilinear group generator is an algorithm $\mathcal{G}$ [8] which takes in a security parameter κ, and outputs five abelian groups G, G_1, U, U_1 and G_T, where $G_1 \subset G$, $U_1 \subset U$. In each group, efficient group operation and random samples are performed. The algorithm outputs efficient computable map $e : G \times U \to G_T$ which has the following properties:

Bilinearity: For any $g_1, g_2 \in G$, $u_1, u_2 \in U$, $e(g_1 g_2, u_1 u_2) = e(g_1, u_1)e(g_1, u_2)e(g_2, u_1)e(g_2, u_2)$;

Non-degenerate: For all $g \in G$, for all $u \in U$, if $e(g, u) = 1$, then $g = 1$.

3.2 Cancelling Pairing

In the proposed scheme, bilinear group generator $\mathcal{G}$ is from the prime order bilinear group generator $\mathcal{P}$, and the pairing e on the product groups is defined as any nontrivial linear combination of the componentwise pairings on the underlying prime order group. In the scheme based on composite order bilinear group [14], if the two group elements g, u have the co-prime order, then $e(g, u) = 1$, which implies that the two subgroups generated by g, u can encode different types of information, and these two components will remain distinct after the pairing operation.

Definition 1. *Let $\mathcal{G}$ be a bilinear group generator [8]. If $\mathcal{G}$ outputs $G_1, \cdots, G_q \subset G$ and $U_1, \cdots, U_q \subset U$, such that:*

1) *$G \cong G_1 \times \cdots \times G_q$ and $U \cong U_1 \times \cdots \times U_q$;*
2) *Whenever $g_i \in G_i$ and $u_j \in U_j$, $i \neq j$, $e(g_i, u_j) = 1$, then $\mathcal{G}$ is q cancelling.*

4 Inner Product Predicate Encryption Scheme with Outsourced Ciphertext Decryption Based on Prime Order Bilinear Group

In this section, we give the syntax and security model of inner product predicate encryption scheme with outsourced ciphertext decryption based on prime order bilinear group.

4.1 Syntax

Let the attribute set be S, and the class of predicate $\mathbb{F}$. Inner product predicate encryption scheme with outsourced ciphertext decryption based on prime order bilinear group comprises the six algorithms as follows:

$Setup(1^k, l) \rightarrow (PK, MS)$**:** The **Setup** algorithm is run by the trusted authority. This algorithm takes in a security parameter 1^κ and a positive integer l, which is attribute and predicate vector length, and outputs the public key PK and the master key MS.

$Encrypt(PK, m, A) \rightarrow CT$**:** The **Encrypt** algorithm is run by the data owner. It takes in the public key PK, message m and attribute A, and outputs the ciphertext $CT = Encrypt(PK, m, A)$.

$PriKeyGen(PK, MS, f) \rightarrow PriKey_f$**:** The private key generation algorithm **PriKeyGen** is run by the trusted authority. It takes in PK, the master key MS and predicate $f \in F \subseteq \mathbb{F}$, and outputs the private key $PriKey_f$.

$OutKeyGen(PK, PriKey_f) \rightarrow (TK_f, SK_f)$**:** The Outsourced Private Key Generation algorithm **OutKeyGen** is run by the data consumer. It takes in the public key PK and the private key $PriKey_f$, and outputs the transformation key TK_f and the private keys SK_f. The transformation key TK_f is public, and the data consumer sends the transformation key TK_f to the cloud server which employs TK_f to partially decrypt the original ciphertext. The data consumer keeps the secret key SK_f private and employs SK_f to decrypt the ciphertext that is partially decrypted. It is the data consumer, not the trusted authority that runs the Outsourced Private Key Generation algorithm **OutKeyGen**, which avoids the costly online request for the trusted authority.

$Transform(PK, TK_f, CT) \rightarrow \widetilde{CT}$**:** The ciphertext **Transform** algorithm is run by the cloud server acting as the proxy. It takes in the public key PK, the transformation key TK_f for the predicate f and the original ciphertext CT, and outputs the partially decrypted ciphertexts $\widetilde{CT}$ if $f(A) = 1$, else the error symbol $\perp$.

$OutDecrypt(PK, SK_f, \widetilde{CT}) \rightarrow \{m, \perp\}$**:** The Outsourced Decryption **OutDecrypt** algorithm is run by the data consumer. It takes in the secret key SK_f and the partially decrypted ciphertexts $\widetilde{CT}$; if $f(A) = 1$, then it outputs the message m, else the error symbol $\perp$.

Correctness. For any security parameters κ, any public keys PK and master secrets MS generated by **Setup**, any $f \in \mathbb{F}$, any private keys

$$\begin{aligned} PriKey_f &\leftarrow \textbf{PriKeyGen}(PK, MS, f), \\ (TK_f, SK_f) &\leftarrow \textbf{OutKeyGen}(PK, PriKey_f), \\ CT &\leftarrow \textbf{Encrypt}(PK, m, A), \\ \widetilde{CT} &\leftarrow \textbf{Transform}(PK, TK_f, CT), \end{aligned}$$

for all attributes $\overrightarrow{y} \in S$, the following propositions hold:

1) If $f(\overrightarrow{y}) = 1$, then **OutDecrypt**($PK$, SK_f, **Transform**(PK, TK_f, $Encrypt(PK, m, A)$)) $\rightarrow m$;
2) If $f(\overrightarrow{y}) = 0$, then **OutDecrypt**($PK$, SK_f, **Transform**(PK, TK_f, $Encrypt(PK, m, A)$)) $\rightarrow \perp$.

4.2 Security Model of Inner Product Predicate Encryption Scheme with Outsourced Ciphertext Decryption Based on Prime Order Bilinear Map

Since security against the adaptive chosen ciphertext attack (**CCA**) requires that any bit of the ciphertext should

not be altered, which makes the requirement too strong, and the outsourced goal is to compress the ciphertext size, therefore, our scheme adopts the replayable adaptive chosen ciphertext attack (**RCCA**) security due to [4]. **RCCA** security allows the ciphertext to be altered provided that the underlying message is not changed in a meaningful way. The security model for inner product predicate encryption scheme with outsourced ciphertext decryption based on prime order bilinear map is described between a challenger and an attacker:

Initialization. The attacker declares the challenge attributes $\overrightarrow{y}, \overrightarrow{z} \in S$, and gives them to the challenger.

Setup. The challenger runs **Setup**$(1^k, l)$ to generate the public key PK and the master secret MS. The challenger defines the value N, and gives it and the public key PK to the attacker.

Query Phase 1. The challenger initializes an empty set T, and an empty set F. The attacker adaptively makes the following queries:

The private key query: On input the predicate f, the challenger runs $PriKey_f \leftarrow$ **PrikeyGen**(PK, MS, f), and sets $F = F \bigcup \{f\}$ with the restriction that $f.\overrightarrow{y} = 0$ holds if and only if $f.\overrightarrow{z} = 0$. It returns $PriKey_f$ to the attacker.

The transformation key query: On input the predicate f, the challenger searches $(f, PriKey_f, TK_f, SK_f)$ to see whether they lie in table T. If so, it returns the transformation key TK_f; else, it runs $PriKey_f \leftarrow$ **PriKeyGen**(PK, MS, f), $(TK_f, SK_f) \leftarrow$ **OutKeyGen**$(PK, PriKey_f)$, and stores $(f, PriKey_f, TK_f, SK_f)$ in table T. It returns the transformation key TK_f to the attacker.

The outsourced decryption query: On input the predicate f, the challenger searches the entry $(f, PriKey_f, TK_f, SK_f)$ to see whether they lies in the table T. If so, it runs $m \leftarrow$ **OutDecrypt**$(PK, SK_f, \widetilde{CT})$, and returns it to the attacker; if not, it returns $\perp$.

Challenge. The attacker submits the two message m_0 and m_1 of the equal length. If $f(\overrightarrow{y}) = f(\overrightarrow{z}) = 1$, then $m_0 = m_1$. The challenger picks a random fairly binary coin $\beta \in \{0,1\}$. If $\beta = 0$, then it gives $CT =$ **Encrypt**$(PK, m_0, \overrightarrow{y})$ to the attacker; else it gives $CT =$ **Encrypt**$(PK, m_1, \overrightarrow{z})$ to the attacker.

Query Phase 2. The same as **Query Phase 1** with the restriction that the attacker cannot:

1) Trivially obtain a private key that decrypt the challenge ciphertext. That is to say, it cannot issue the query that the attributes associated with the ciphertext satisfy the predicate associated with the private key.
2) Trivially issue decryption query. The decryption query is the same as the **Query Phase 1**.

Guess. The attacker outputs a guess β' of β.

In this game, the advantage of the attacker is defined as

$$Adv_{\mathcal{A}} = |Pr\{\beta' = \beta\} - \frac{1}{2}|.$$

Definition 2. *If all probabilistic polynomial time attackers have the negligible advantage in the aforementioned **RCCA** security games, then inner product predicate encryption scheme with outsourced ciphertext decryption based on prime order bilinear group is **RCCA** secure.*

CPA Security. If decryption oracles in **Query Phase 1** and **Query Phase 2** are removed, then this scheme is secure against chosen plaintext attack **CPA**.

Selective Security. An inner product predicate encryption scheme with outsourced ciphertext decryption based on prime order bilinear group is selectively secure if an Init stage is added before *Setup*, in which the attacker declares the challenge attributes.

5 The Architecture of Inner Product Predicate Encryption Scheme with Outsourced Ciphertexts Decryption Based on Prime Order Bilinear Map

We proposed an inner product predicate encryption scheme with outsourced ciphertexts decryption based on prime order bilinear map whose architecture is illustrated in Figure 1. In the architecture, the cloud server stores the inner product predicate encryption (IPPE) ciphertext CT; when the client employed by the data consumer attempts to decrypt the ciphertext CT, if the ciphertext CT is found that it is not partially decrypted, then it sends the ciphertext CT and the transformation key TK_f to the cloud server which employs the transformation key TK_f to run the **Transform** algorithm to output the partially decrypted ciphertext $\widetilde{CT}$. The cloud server sends $\widetilde{CT}$ to the client which employs the secret key SK_f to decrypt the partially decrypted ciphertext $\widetilde{CT}$ to obtain the plaintext message.

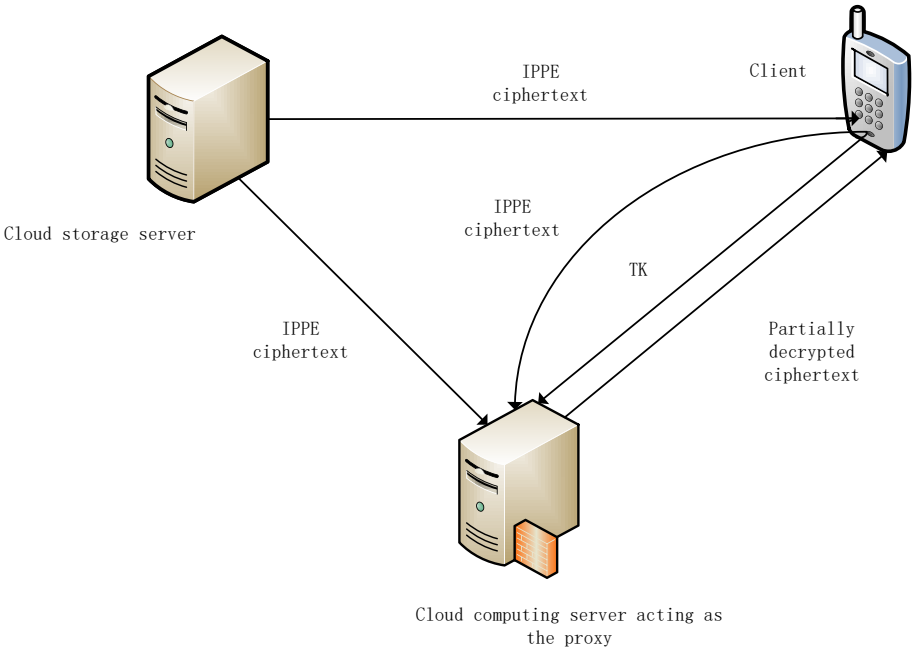


Figure 1: Architecture of inner product predicate encryption scheme with outsourced ciphertexts decryption

6 The Construction of Outsourced Ciphertext Decryption of Inner Product Predicate Encryption Scheme Based on Prime Order Bilinear Map

Our scheme employs the asymmetric bilinear pairing generator to improve the efficiency.

Setup$(1^k, l)$**:** The **Setup** algorithm is run by the trusted authority. It takes in a security parameter 1^κ and a positive integer l that is attribute and predicate vector length,$\mathcal{G}(\kappa)$ is a 3-cancelling bilinear group generator, and outputs the groups G_i and U_i which have the prime order exponents p_i respectively, where $i = 1, \cdots, 3$. Define the hash functions as follows: $H_1 : G_T \times \{0,1\}^n \to \mathbb{Z}_{p_1}$, $H_2 : G_T \to \{0,1\}^n (n$ is the length of message bit), and $H_3 : \{0,1\}^n \to \mathbb{Z}_{p_1}$. Perform the following:

Step 1: Calculate $(G, G_1, G_2, G_3, U, U_1, U_2, U_3, G_T) \xleftarrow{\$} \mathcal{G}(\kappa)$.

Step 2: Pick $g_i \xleftarrow{\$} G_i$ and $u_i \xleftarrow{\$} U_i$.

Step 3: Pick $\mu \xleftarrow{\$} \mathbb{Z}_{p_1}$ and $t_0 \xleftarrow{\$} \mathbb{Z}_{p_3}$.

Step 4: For $j = 1, \cdots, l$, pick $\eta_{1,j}, \eta_{2,j} \xleftarrow{\$} \mathbb{Z}_{p_1}$ and $\sigma_{1,j}, \sigma_{2,j} \xleftarrow{\$} \mathbb{Z}_{p_3}$.

Step 5: Output the public parameters $(g_1, g_3, W = g_2 g_3^{t_0}, e(g_1, u_1)^\mu, \{B_{1,j} = g_1^{\eta_{1,j}} g_3^{\sigma_{1,j}}, B_{2,j} = g_1^{\eta_{2,j}} g_3^{\sigma_{2,j}}\}_{j=1}^l)$.

The master secret $MS = (u_1, u_2, u_3, u_1^{-\mu}, \{\eta_{1,j}, \eta_{2,j}\}_{j=1}^l)$.

Encrypt$(PK, m, \overrightarrow{y})$**:** Let $\overrightarrow{y} = (y_1, \cdots, y_l) \in \mathbb{Z}_N^l$ is an attribute vector, $m \in \{0,1\}^n$ is a message whose length is n. The **Encrypt** algorithm picks a random $\gamma \in G_T$, calculates $s = H_1(\gamma, m)$, $d = H_2(\gamma)$ and $\tau = H_3(d)$, and picks $\delta, \theta \xleftarrow{\$} \mathbb{Z}_N$ and random values $\xi_{1,j}, \xi_{2,j} \xleftarrow{\$} \mathbb{Z}_N$, where $j = 1, \cdots, l$. The ciphertext CT is published as:

$$CT = (E', E_b, \widehat{E}, \tau, \{E_{1,j}, E_{2,j}\}_{j=1}^l).$$

Here,

$$\begin{cases} E' = \gamma e(g_1, u_1)^{\mu s}, \\ E_b = g_1^s, \\ \widehat{E} = m \oplus d, \\ \tau = H_3(d), \\ E_{1,j} = B_{1,j}^s W^{\delta y_j} g_3^{\xi_{1,j}}, \\ E_{2,j} = B_{2,j}^s W^{\theta y_j} g_3^{\xi_{2,j}} \end{cases}$$

PriKeyGen(PK, MS, f)**:** Let $f = (f_1, \cdots, f_l) \in \mathbb{Z}_N^l$ is the predicate vector. The **PriKeyGen** algorithm picks $t_{1,j}, t_{2,j} \in \mathbb{Z}_N$, where $j = 1, \cdots, l$, and $\omega_1, \omega_2, \psi_1, \psi_2 \in \mathbb{Z}_N$. It outputs the private key corresponding to f:

$$PriKey_f = (K_b, \{K_{1,j}, K_{2,j}\}_{j=1}^l).$$

Here,

$$\begin{cases} K_b &= u_1^{-\mu} u_2^{\psi_1} u_3^{\psi_2} \prod_{j=1}^{l} u_1^{-t_{1,j}\eta_{1,j} - t_{2,j}\eta_{2,j}} \\ K_{1,j} &= u_1^{t_{1,j}} u_2^{\omega_1 f_j} \\ K_{2,j} &= u_1^{t_{2,j}} u_2^{\omega_2 f_j} \end{cases}$$

$OutKeyGen(PK, PriKey_f)$: The Outsourced Private Key Generation **OutKeyGen** algorithm runs **PriKeyGen**(PK, MS, f) algorithm to obtain $\widetilde{PriKey_f} = (\widetilde{K_b} = u_1^{-\mu} u_2^{\psi_1} u_3^{\psi_2} \prod_{j=1}^{l} u_1^{-t_{1,j}\eta_{1,j} - t_{2,j}\eta_{2,j}}, \{\widetilde{K_{1,j}} = u_1^{t_{1,j}} u_2^{\omega_1 f_j}, \widetilde{K_{2,j}} = u_1^{t_{2,j}} u_2^{\omega_2 f_j}\}_{j=1}^{l})$. It picks a random $x \in \mathbb{Z}_{p_1}^*$, and it sets the transformation key TK_f as:

$$\begin{aligned} TK_f &= (PK, OK_b, OK_{1,j}, OK_{2,j}). \\ OK_b &= \widetilde{K_b}^{\frac{1}{x}}, \\ OK_{1,j} &= \widetilde{K_{1,j}}^{\frac{1}{x}}, \\ OK_{2,j} &= \widetilde{K_{2,j}}^{\frac{1}{x}}. \end{aligned}$$

The private key $DecryptKey$ is $(x, TK_f) = (SK_f, TK_f)$.

$Transform(PK, TK_f, CT)$**:** The **Transform** algorithm takes in TK_f associated with the predicate f and the ciphertext CT associated with the attributes. If the attributes $\overrightarrow{y}$ associated with the ciphertext do not satisfy the predicate f, then the **Transform** algorithm returns $\perp$, else it calculates:

$$\begin{aligned} & e(E_b, OK_b) \prod_{j=1}^{l} e(E_{1,j}, OK_{1,j}) e(E_{2,j}, OK_{2,j}) \\ = \; & e(g_1^s, (u_1^{-\mu} u_2^{\psi_1} u_3^{\psi_2} \prod_{j=1}^{l} u_1^{-t_{1,j}\eta_{1,j} - t_{2,j}\eta_{2,j}})^{\frac{1}{x}}) \\ & \prod_{j=1}^{l} e(B_{1,j}^s W^{\delta y_j} g_3^{\xi_{1,j}}, (u_1^{t_{1,j}} u_2^{\omega_1 f_j})^{\frac{1}{x}}) \\ & e(B_{2,j}^s W^{\theta y_j} g_3^{\xi_{2,j}}, (u_1^{t_{2,j}} u_2^{\omega_2 f_j})^{\frac{1}{x}}) \\ = \; & e(g_1^s, u_1^{-\mu})^{\frac{1}{x}} e(g_1^s, (\prod_{j=1}^{l} u_1^{-t_{1,j}\eta_{1,j} - t_{2,j}\eta_{2,j}})^{\frac{1}{x}}) \\ & \prod_{j=1}^{l} e(g_1^{s\eta_{1,j}} g_2^{\delta y_j}, (u_1^{t_{1,j}} u_2^{\omega_1 f_j})^{\frac{1}{x}}) \\ & e(g_1^{s\eta_{2,j}} g_2^{\theta y_j}, (u_1^{t_{2,j}} u_2^{\omega_2 f_j})^{\frac{1}{x}}) \\ = \; & e(g_1^s, u_1^{-\mu})^{\frac{1}{x}} e(g_1^s, (\prod_{j=1}^{l} u_1^{-t_{1,j}\eta_{1,j} - t_{2,j}\eta_{2,j}})^{\frac{1}{x}}) \\ & \prod_{j=1}^{l} e(g_1^{s\eta_{1,j}} g_2^{\delta y_j}, (u_1^{t_{1,j}} u_2^{\omega_1 f_j})^{\frac{1}{x}}) \\ & e(g_1^{s\eta_{2,j}} g_2^{\theta y_j}, (u_1^{t_{2,j}} u_2^{\omega_2 f_j})^{\frac{1}{x}}) \\ = \; & e(g_1^s, u_1^{-\mu})^{\frac{1}{x}} e(g_1^s, (\prod_{j=1}^{l} u_1^{-t_{1,j}\eta_{1,j} - t_{2,j}\eta_{2,j}})^{\frac{1}{x}}) \\ & e(g_1^s, (\prod_{j=1}^{l} u_1^{t_{1,j}\eta_{1,j} + t_{2,j}\eta_{2,j}})^{\frac{1}{x}}) \\ & \prod_{j=1}^{l} e(g_2^{\delta y_j}, (u_2^{\omega_1 f_j})^{\frac{1}{x}}) e(g_2^{\theta y_j}, (u_2^{\omega_2 f_j})^{\frac{1}{x}}) \\ = \; & e(g_1^s, u_1^{-\mu})^{\frac{1}{x}} \prod_{j=1}^{l} e(g_2^{\delta y_j}, (u_2^{\omega_1 f_j})^{\frac{1}{x}}) e(g_2^{\theta y_j}, (u_2^{\omega_2 f_j})^{\frac{1}{x}}) \\ = \; & e(g_1, u_1)^{\frac{-\mu s}{x}} \prod_{j=1}^{l} e(g_2, u_2)^{\frac{1}{x}\delta y_j \omega_1 f_j} e(g_2, u_2)^{\frac{1}{x}\theta y_j \omega_2 f_j}) \\ = \; & e(g_1, u_1)^{\frac{-\mu s}{x}} \prod_{j=1}^{l} e(g_2, u_2)^{\frac{1}{x}\delta \omega_1 y_j f_j} e(g_2, u_2)^{\frac{1}{x}\theta \omega_2 y_j f_j}) \\ = \; & e(g_1, u_1)^{\frac{-\mu s}{x}} \prod_{j=1}^{l} e(g_2, u_2)^{\frac{1}{x}(\delta\omega_1 + \theta\omega_2)(y_j f_j)} \\ = \; & e(g_1, u_1)^{\frac{-\mu s}{x}} e(g_2, u_2)^{\frac{1}{x}(\delta\omega_1 + \theta\omega_2) \sum_{j=1}^{l} < y_j, f_j >} \\ = \; & e(g_1, u_1)^{\frac{-\mu s}{x}} \; (\text{If} < \overrightarrow{y}, f >= 0). \end{aligned}$$

If $< \overrightarrow{y}, f >= 0$, then the **Transform** algorithm outputs the result as $= e(g_1, u_1)^{\frac{-\mu s}{x}}$; else it returns the error symbol $\perp$.

The **Transform** algorithm outputs the partially decrypted ciphertext:

$$\begin{aligned} \widetilde{CT} &= (E', \widehat{E}, \tau, E_e). \\ \widehat{E} &= m \oplus d, \\ \tau &= H_3(d), \\ E_e &= e(g_1, u_1)^{\frac{-\mu s}{x}}. \end{aligned}$$

OutDecrypt$(PK, DecryptKey, \widetilde{CT}) \rightarrow \{m, \perp\}$. The Outsourced Decryption **OutDecrypt** algorithm takes in the public key PK, the private key $DecryptKey = (x, TK_f)$ and the partially decrypted ciphertext $\widetilde{CT}$. If the ciphertext is not partially decrypted, then **OutDecrypt** algorithm first runs the **Transform**(PK, TK_f, CT) algorithm. If the **Transform**(PK, TK_f, CT) algorithm outputs $\perp$, then **OutDecrypt**$(PK, DecryptKey, \widetilde{CT})$ algorithm outputs $\perp$, else it takes in $\widetilde{CT}$ and calculates

$$\begin{aligned} \gamma &= E' E_e^x, \\ d &= H_2(\gamma), \\ m &= \widehat{E} \oplus H_2(\gamma), \\ \tau' &= H_3(d). \end{aligned}$$

This algorithm checks that $\tau' \stackrel{?}{=} \tau$. If so, it will show that the transformation done by the server is correct. It outputs the message m.

If the ciphertext is partially decrypted by the cloud server, then **OutDecrypt** algorithm requires one exponentiation, one hash operation and XOR operation to obtain the message m, and no pairing operation. Since the malicious cloud server cannot obtain the message, our scheme is secure against it. Through the hash function H_3, the data consumer can decide whether the transformation done by the cloud server is correct.

7 Proof of Security

Suppose there is a **PPT** attacker $\mathcal{A}$ that attacks the proposed scheme in the selective **RCCA** security model with the advantage ϵ. We build a simulator $\mathcal{B}$ which attacks the [8] scheme in the selective **CPA** security model with the advantage ϵ. Freeman scheme [8] is proven secure under the two assumptions.

Initialization. The simulator $\mathcal{B}$ runs the attacker $\mathcal{A}$ which declares the challenge attributes $(\overrightarrow{y}^*, \overrightarrow{z}^*)$ that the simulator $\mathcal{B}$ sends to the Freeman scheme [8] challenger as the challenge attributes on which it wished to be challenged.

Setup. The simulator $\mathcal{B}$ obtains the Freeman [8] public parameters which are sent to the attacker $\mathcal{A}$ as the public parameters.

Query Phase 1. The simulator $\mathcal{B}$ initializes empty tables T, T_1, T_2, T_3 and an empty set F. The adversary's queries are answered by the simulator $\mathcal{B}$ as follows.

Random Oracle Hash $H_1 : G_T \times \{0,1\}^n \to \mathbb{Z}_{p_1}$**:** If there is an entry (γ, m, s) in the table T_1, then it returns s, else, it picks a random value $s \in \mathbb{Z}_p$, records (γ, m, s) in table T_1, and returns s.

Random Oracle Hash $H_2 : G_T \to \{0,1\}^n$**:** If there is an entry (γ, d) in table T_2, then it returns d, else it picks a random value $d \in \{0,1\}^n$, records (γ, d) in table T_2, and returns d.

Random Oracle Hash $H_3 : \{0,1\}^n \to \mathbb{Z}_{p_1}$**:** If there is an entry (τ, d) in table T_3, then it returns τ, else it picks a random value $d \in \{0,1\}^n$, records (τ, d) in table T_3, and returns τ.

The simulator $\mathcal{B}$ proceeds as follows: If the challenge attributes satisfy the predicate, the transformation key is constructed as follows: Call the private key generation algorithm of the Freeman scheme [8] to obtain the private key associated with the predicate f as $(\widetilde{K_b}, \widetilde{K_{1,j}}, \widetilde{K_{2,j}})$. The algorithm picks a random value $x \in \mathbb{Z}^*_{p_1}$, sets the transformation key as $TK_f = (PK, OK_b = \widetilde{K_b}^{\frac{1}{x}}, OK_{1,j} = \widetilde{K_{1,j}}^{\frac{1}{x}}, OK_{2,j} = \widetilde{K_{2,j}}^{\frac{1}{x}})$, stores it in table T, and returns TK_f to the attacker, else it returns $\perp$.

The attacker cannot issue the private key query where the ciphertext attributes satisfy the predicate. If there is an entry $(f, PriKey_f, TK_f, SK_f)$ in table T, then the simulator $\mathcal{B}$ obtains $(f, PriKey_f, TK_f, SK_f)$, set $F = F \bigcup \{f\}$, and returns the private key $PriKey_f$ to the attacker, else it returns $\perp$.

Decryption Oracle: Assume that all ciphertext inputs to the oracle are the partially decrypted ciphertext. The simulator $\mathcal{B}$ and the attacker $\mathcal{A}$ have access to the transformation key TK_f, so they can execute the transformation operation.Let $CT = (E' = \gamma e(g_1, u_1)^{\mu s}, E_b = g_1^s, \widehat{E} = m \oplus d, \tau = H_3(d), \{E_{1,j} = B_{1,j}^s W^{\delta y_j} g_3^{\xi_{1,j}}, E_{2,j} = B_{2,j}^s W^{\theta y_j} g_3^{\xi_{2,j}}\}_{j=1}^l)$ be associated with the attribute vector $\overrightarrow{y}$. From table T, $(f, PriKey_f, TK_f, SK_f)$ is obtained. If no entry exists or the attributes associated with the ciphertext do not satisfy the predicate, then it returns $\perp$ to the attacker $\mathcal{A}$.

If the attributes associated with the ciphertext do satisfy the predicate, proceed as follows: Parse $PriKey_f = (SK_f, TK_f) = (x, TK_f)$, and calculate $\gamma = E' \backslash E_e^x$.

Test if $E' = \gamma e(g_1, u_1)^{\mu s}, \widehat{E} = m \oplus d$. If so, it outputs the message m, else it returns $\perp$ to the attacker.

Challenge. The attacker $\mathcal{A}$ submits the two messages m_0^*, m_1^* of equal length, and the simulator $\mathcal{B}$ proceeds as follows:

Step 1. The simulator $\mathcal{B}$ picks random messages $(\gamma_0, \gamma_1) \in G_T$, and passes them to the challenger in Freeman scheme [8] to obtain the ciphertext $CT = (E' = \gamma e(g_1, u_1)^{\mu s}, E_b = g_1^s, \widehat{E} = m \oplus d, \tau = H_3(d), \{E_{1,j} = B_{1,j}^s W^{\delta y_j} g_3^{\xi_{1,j}}, E_{2,j} = B_{2,j}^s W^{\theta y_j} g_3^{\xi_{2,j}}\}_{j=1}^l)$.

Step 2. The simulator $\mathcal{B}$ picks the random value $\widehat{E}' \in \{0,1\}^n$.

Step 3. The simulator $\mathcal{B}$ sends to the attacker $\mathcal{A}$ the challenger ciphertext $CT^* = (E', E_b, \widehat{E}', \tau = H_3(d), \{E_{1,j}, E_{2,j}\}_{j=1}^l)$.

Query Phase 2. The simulator $\mathcal{B}$ continues to answer the queries as **Query Phase 1**, except that if the response to the decryption query is m_0^*,m_1^*, then the simulator $\mathcal{B}$ responds with `test`.

Guess. The attacker $\mathcal{A}$ returns a bit β or abort, and the simulator $\mathcal{B}$ does not respond. The simulator $\mathcal{B}$ search through tables T_1 and T_2 to see if γ_0 or γ_1 appears as the first element of any entry. If both γ_0 and γ_1 or neither appear, the simulator $\mathcal{B}$ returns a random bit as a guess. If γ_β appears, then the simulator $\mathcal{B}$ returns β as its guess.

Table 1: Comparison of our scheme and freeman scheme

References	**Security Level**	**Original Ciphertext Size**	**Time to Decrypt the Original Ciphertext**	**Outsourced Ciphertext Size**	**Time to Decrypt the Outsourced Ciphertext**
Freeman Scheme [8]	CPA security	$\|G_T\| + \|G_1\| + \|G_1\|l\|G_2\|(l+1)\|G_3\|$	$\leq (1+2l)C_e$	Not Supported	Not Supported
Our Scheme	**RCCA** security	$\|G_T\| + \|G_1\| + \|G_1\|l\|G_2\|(l+1)\|G_3\| + n + 1H_2 + 1H_3$	$\leq (1+2l)C_e$	$\|G_T\| + n + 1H_2 + 1H_3$	E_T+$1H_2$+$1H_3$

8 Performance Evaluation

As illustrated in Table 1 which depicts the outsourced ciphertext decryption of inner product predicate encryption scheme based on prime order bilinear group, where l denotes the attribute and predicate vector length, n is the length of message bits in **RCCA** scheme, aH_2, bH_3 denotes a times H_2 operation,b times H_3 operation, or the bit number of hash operation, respectively, and cC_e, $||$, and E_T denotes c times bilinear map, the cardinality of the set, and exponentiation. As seen from the Table 1, in contrast with Freeman scheme [8], in the proposed scheme, the data consumer requires E_T+$1H_2$+$1H_3$ operation, outsourcing significantly reduces the time to decrypt for the data consumer, and compresses the ciphertext size, such that the overhead of the data consumer is significantly reduced. Furthermore, our scheme achieves security against **RCCA**, whereas Freeman scheme [8] is only secure against **CPA**.

9 Conclusion

In predicate encryption scheme, the ciphertext size and time to decrypt it scale with the complexity of the predicate. If predicate encryption scheme is employed in the resource constrained devices to achieve fine grained access control over the encrypted data, it will drain the battery. In this work, we propose inner product predicate encryption scheme based on prime order bilinear group that outsources decryption of ciphertext to the cloud server, which significantly reduces the ciphertext size and time to decrypt it, whereas the cloud server does not learn the underlying plaintext message. Therefore, outsourced decryption has the obvious advantages.

10 Acknowledgments

This work was supported by national science and technology support program of China (No.2014BAH11F02), the National Science Foundation of China (No. 61402376 and No.61373163) and the science and technology foundation of Sichuan Province of China (No. 2014GZ0109). The authors gratefully acknowledge Chaosheng Feng and the anonymous reviewers for their valuable comments.

References

[1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[2] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology (EUROCRYPT'04)*, LNCS 3027, pp. 506–522, 2004.

[3] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Theory of Cryptography Conference (TCC'07)*, pp. 535–554, Springer, 2007.

[4] R. Canetti, H. Krawczyk, and J. B. Nielsen, "Relaxing chosen-ciphertext security," in *Advances in Cryptology (CRYPTO'03)*, pp. 565–582, Springer, 2003.

[5] B. Chevallier-Mames, J. Coron, N. McCullagh, D. Naccache, and M. Scott, "Secure delegation of elliptic-curve pairing," in *International Conference on Smart Card Research and Advanced Applications*, pp. 24–35, Springer, 2010.

[6] K. M. Chung, Y. Kalai, and S. P. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Advances in Cryptology (CRYPTO'10)*, pp. 483–501, Springer, 2010.

[7] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *International Journal of Network Security*, vol. 16, no. 1, pp. 1–13, Jan. 2014.

[8] D. M. Freeman, "Converting pairing-based cryptosystems from composite order groups to prime-order groups," in *Advances in Cryptology (EUROCRYPT'10)*, pp. 44–61, Springer, 2010.

[9] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Advances in Cryptology (CRYPTO'10)*, pp. 465–482, Springer, 2010.

[10] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing (STOC'09)*, pp. 169–178, 2009.
[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06)*, pp. 89–98, 2006.
[12] M. Green, S. Hohenberger, and B.Waters, "Outsourcing the decryption of abe ciphertexts," in *USENIX Security Symposium*, pp. 354–372, 2011.
[13] V. Iovino and G. Persiano, "Hidden-vector encryption with groups of prime order," in *Pairing-Based Cryptography (Pairing'08)*, LNCS 5209, pp. 75–88, Springer, 2008.
[14] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunction, polynomial equations, and inner products," in *Advances in Cryptology (EUROCRYPT'08)*, pp. 146–162, Springer, 2008.
[15] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal of Network Security*, vol. 15, no. 4, pp. 231–240, July 2013.
[16] B. Parno, M. Raykova, and V. Vaikuntanathan, "How to delegate and verify in public: Verifiable computation from attribute-based encryption," in *Theory of Cryptography (TCC'12)*, LNCS 7194, pp. 422–439, Springer, 2012.
[17] M. Scott., *Personal Communication*, 2009.
[18] E. Shi and B. Waters, "Delegating capabilities in predicate encryption systems," in *International Colloquium on Automata, Languages, and Programming (ICALP'08)*, pp. 560–578, Springer, 2008.

Xingbing Fu is a lecturer, he received his M.S. degree from Southwest University in 2007. He is currently a PhD Candidate School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interests are information security, cloud computing, and cryptography.

Xuyun Nie received the Ph.D. degree in Information security from Graduate university of Chinese Academy of Sciences, Beijing, China, in 2007. He is presently an Associate Professor at University of Electronic Science and Technology of China (UESTC). His research interests include cryptography and information security.

Fagen Li received the Ph.D. degree in Cryptography from Xidian University, Xi'an, P.R. China in 2007. His research interests include cryptography and network security.

A Note on Two Schemes for Secure Outsourcing of Linear Programming

Zhengjun Cao[1], Lihua Liu[2]
(Corresponding author: Lihua Liu)
Department of Mathematics, Shanghai University[1]
No.99, Shangda Road, Shanghai, China.
Department of Mathematics, Shanghai Maritime University[2]
No. 1550, Haigang Ave, Pudong New District, Shanghai, China.
(Email: liulh@shmtu.edu.cn)
(Received Jan. 26, 2016; revised and accepted Apr. 23 & May 31, 2016)

Abstract

Recently, Wang et al. [IEEE INFOCOM 2011, 820-828], and Nie et al. [IEEE AINA 2014, 591-596] have proposed two schemes for secure outsourcing of linear programming (LP). They did not consider the standard form: minimize $\mathbf{c}^T\mathbf{x}$, subject to $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0$. Instead, they studied a peculiar form: minimize $\mathbf{c}^T\mathbf{x}$, subject to $\mathbf{Ax} = \mathbf{b}, \mathbf{Bx} \geq 0$, where $\mathbf{B}$ is a non-singular matrix. In this note, we stress that the proposed peculiar form is unsolvable and meaningless. The two schemes have confused the *functional inequality constraints* $\mathbf{Bx} \geq 0$ with the *nonnegativity constraints* $\mathbf{x} \geq 0$ in the linear programming model. But the condition $\mathbf{x} \geq 0$ is indispensable to LP. Thus, both two schemes failed.

Keywords: Cloud computing, functional inequality constraints, linear programming, nonnegativity constraints, simplex method.

1 Introduction

Cloud computing makes use of the massive resources of computing and storage systems via the Internet to efficiently deal with information processing. It supports a paradigm shift from local to network-centric computing and network-centric content [10, 17], and benefits scientific and engineering applications, such as data mining, computational financing, and many other computational and data-intensive activities [14, 18]. Cloud computing makes it possible to enable customers with limited computational resources to outsource large-scale computational tasks to the cloud, including linear equations (LE), linear programming (LP), matrix multiplication computation, and matrix inversion computation.

In 2011, Dreier and Kerschbaum [4] put forth a method for secure outsourcing of LP. In order to protect the solution $\mathbf{x}$, the Dreier-Kerschbaum scheme uses the affine transformation

$$\mathbf{z} = \mathbf{Q}^{-1}\mathbf{x} + \mathbf{r},$$

where $\mathbf{Q}$ is a positive monomial matrix (a monomial matrix contains exactly one non-zero entry per row and column), and $\mathbf{r}$ is a random vector picked by the client. Wang et al. [15] also presented a scheme for outsourcing of LP based on the transformation $\mathbf{y} = \mathbf{M}^{-1}(\mathbf{x} + \mathbf{r})$, where $\mathbf{M}$ is a random non-singular matrix and $\mathbf{r}$ is a random vector. In 2014, Nie et al. [11] proposed another scheme for outsourcing of LP based on the same transformation as that used in [15].

In 2013, Lei et al. [8] have proposed a scheme for outsourcing matrix inversion computation over the field $\mathbb{R}$ of real numbers. After that, they [7] proposed another scheme for outsourcing matrix multiplication computation over $\mathbb{R}$. But the verifying equations in [7, 8] do not hold over $\mathbb{R}$ because the computational errors, especially rounding errors, are not considered carefully. That means the client cannot check whether the cloud server is cheating him.

Wang et al. [16] have ever proposed a scheme for outsourcing large-scale systems of linear equations to cloud, which enables a client to securely harness the cloud for iteratively finding successive approximations to the LE solution, while keeping both the sensitive input and output of the computation private. Recently, Cao and Liu [1] pointed out that the Wang et al.'s scheme fails because the involved homomorphic encryption system [2, 12] is invalid in the context of the scheme. In 2014, Chen et al. [3] proposed two computation outsourcing schemes for LE and LP. Both two schemes are insecure because the technique of masking a vector with a diagonal matrix is vulnerable to statistical analysis attacks. In 2015, Salinas et al. [13] proposed a scheme for outsourcing LE, which makes use of the conjugate gradient method to solve the equivalent quadratic program in the client-server scenario. Recently,

Hsien et al. [6, 9] presented two surveys of public auditing for secure data storage in cloud computing.

In this note we would like to stress that the proposed peculiar form by Wang et al. [15] and Nie et al. [11] is unsolvable and meaningless. In fact, they did not consider the standard form:

$$\text{Minimize } \mathbf{c}^T\mathbf{x}, \text{ subject to } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0.$$

Instead, they studied a peculiar form:

$$\text{Minimize } \mathbf{c}^T\mathbf{x}, \text{ subject to } \mathbf{Ax} = \mathbf{b}, \mathbf{Bx} \geq 0,$$

where $\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{c}$ is an $n \times 1$ vector, $\mathbf{b}$ is an $m \times 1$ vector, $\mathbf{x}$ is an $n \times 1$ vector of variables, and $\mathbf{B}$ is an $n \times n$ non-singular matrix.

They have confused the *functional inequality constraints* $\mathbf{Bx} \geq 0$ with *nonnegativity constraints* $\mathbf{x} \geq 0$ in the linear programming model. In nature, the condition $\mathbf{x} \geq 0$ is indispensable to LP. Thus, both two schemes failed. We also review the possible method for secure outsourcing of LP, which is due to Dreier and Kerschbaum.

2 Preliminaries

Linear programming has numerous important applications. Among these allocating resources to activities is the most common type of application. The *standard form* for a linear programming problem can be described as follows [5]. Select the values for $x_1, \cdots, x_n$ so as to

$$\text{maximize} \quad c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

subject to the restrictions

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m \end{aligned}$$

and

$$x_1 \geq 0, x_2 \geq 0, \cdots, x_n \geq 0.$$

$c_1x_1 + c_2x_2 + \cdots + c_nx_n$ is called the *objective function*. The first m constraints are sometimes called *functional constraints*. The restrictions $x_j \geq 0$ are called *nonnegativity constraints*.

The simplex method, a general procedure for solving linear programming problems, is based on solving systems of equations. Therefore, it has to firstly convert the functional inequality constraints to *equivalent equality constraints*. This conversion is accomplished by introducing *slack variables*. After the conversion, the original linear programming model can now be replaced by the equivalent model (called the *augmented form*).

Using matrices, the standard form for the general linear programming model becomes

$$\text{maximize } \mathbf{c}^T\mathbf{x}, \text{ subject to } \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0$$

where $\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{c}$ is an $n \times 1$ vector, $\mathbf{b}$ is an $m \times 1$ vector, and $\mathbf{x}$ is an $n \times 1$ vector of variables. To obtain the augmented form of the problem, introduce the column vector of slack variables $\mathbf{x}_s = (x_{n+1}, \cdots, x_{n+m})^T$ so that the constraints become

$$[\mathbf{A}, \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \mathbf{b} \text{ and } \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} \geq \mathbf{0},$$

where $\mathbf{I}$ is the $m \times m$ identity matrix, and the null vector $\mathbf{0}$ now has $n + m$ elements.

Notice that the nonnegativity constraints are left as inequalities because they are used to determine the *leaving basic variable* according to the *minimum ratio test*.

3 Analysis of Two Schemes for Outsourcing of LP

3.1 Review

We now take the scheme in [15] as the example to show the incorrectness of the proposed peculiar form (see page 822 of [15] and page 592 of [11]). In the scheme, there are two entities, the client and the cloud server. The client has the original problem

$$\min \mathbf{c}^T\mathbf{x}, \text{ s.t. } \mathbf{Ax} = \mathbf{b}, \ \mathbf{Bx} \geq 0 \tag{1}$$

where $\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{c}$ is an $n \times 1$ vector, $\mathbf{b}$ is an $m \times 1$ vector, $\mathbf{x}$ is an $n \times 1$ vector of variables, $\mathbf{B}$ is an $n \times n$ non-singular matrix.

To ensure the privacy of input and output, the client transforms the original problem into the following problem

$$\min \mathbf{c}'^T\mathbf{y}, \text{ s.t. } \mathbf{A}'\mathbf{y} = \mathbf{b}', \ \mathbf{B}'\mathbf{y} \geq 0 \tag{2}$$

where

$$\begin{cases} \mathbf{A}' = \mathbf{QAM} \\ \mathbf{B}' = (\mathbf{B} - \mathbf{PQA})\mathbf{M} \\ \mathbf{b}' = \mathbf{Q}(\mathbf{b} + \mathbf{Ar}) \\ \mathbf{c}' = \gamma\mathbf{M}^T\mathbf{c} \\ \mathbf{y} = \mathbf{M}^{-1}(\mathbf{x} + \mathbf{r}) \end{cases}$$

satisfying

$$|\mathbf{B}'| \neq 0, \mathbf{Pb}' = \mathbf{Br}, \mathbf{b} + \mathbf{Ar} \neq 0, \gamma > 0,$$

where $\mathbf{P}$ is an $n \times m$ matrix, $\mathbf{Q}$ is a random $m \times m$ non-singular matrix, $\mathbf{M}$ is a random $n \times n$ non-singular matrix, and $\mathbf{r}$ is an $n \times 1$ vector. The client then sends Problem (2) to the server.

3.2 Analysis

Upon receiving Problem (2), the server has to introduce the nonnegativity conditions $\mathbf{y} \geq 0$ into it and solve the following problem

$$\min \mathbf{c}'^T\mathbf{y}, \text{ s.t. } \mathbf{A}'\mathbf{y} = \mathbf{b}', \ \mathbf{B}'\mathbf{y} \geq 0, \ \mathbf{y} \geq 0 \tag{3}$$

This is because the constraints $\mathbf{B}'\mathbf{y} \geq 0$ should be viewed as a part of the functional constraints, not the necessary nonnegativity constraints, unless

$$\mathbf{B}' = (\mathbf{B} - \mathbf{PQA})\mathbf{M}$$

can be rewritten as a diagonal matrix where the entries on the main diagonal are strictly positive (in such case, $\mathbf{B}'\mathbf{y} \geq 0$ implies $\mathbf{y} \geq 0$).

Unfortunately, the solution of the following problem

$$\min \mathbf{c}^T\mathbf{x}, \text{ s.t. } \mathbf{Ax} = \mathbf{b}, \ \mathbf{Bx} \geq 0, \ \mathbf{x} \geq 0 \qquad (4)$$

cannot be derived from the solution of Problem (3), because the transformation

$$\mathbf{y} = \mathbf{M}^{-1}(\mathbf{x} + \mathbf{r}), \text{ where } \mathbf{x} \geq 0$$

cannot ensure that $\mathbf{y} \geq 0$.

The authors of [11, 15] have confused the functional inequality constraints $\mathbf{Bx} \geq 0$ with the nonnegativity constraints $\mathbf{x} \geq 0$. In fact, the proposed peculiar form is meaningless and unsolvable, unless $\mathbf{Bx} \geq 0$ can be rewritten as $\mathbf{x} \geq 0$.

4 A Possible Method for Secure Outsourcing of LP

In 2011, Dreier and Kerschbaum [4] have already presented a possible method for secure outsourcing of LP. The scheme can be briefly described as follows.

Given the original LP problem

$$\min \mathbf{c}^T\mathbf{x}, \text{ s.t. } \mathbf{M}_1\mathbf{x} = \mathbf{b}_1, \mathbf{M}_2\mathbf{x} \leq \mathbf{b}_2, \mathbf{x} \geq 0,$$

the client uses *a positive monomial matrix* $\boldsymbol{Q}$ (a monomial matrix contains exactly one non-zero entry per row and column) to hide $\mathbf{c}$ and obtains

$$\begin{aligned} & \min \mathbf{c}^T\mathbf{Q}\mathbf{Q}^{-1}\mathbf{x}, \\ \text{s.t.} \quad & \mathbf{M}_1\mathbf{Q}\mathbf{Q}^{-1}\mathbf{x} = \mathbf{b}_1, \\ & \mathbf{M}_2\mathbf{Q}\mathbf{Q}^{-1}\mathbf{x} \leq \mathbf{b}_2, \\ & \mathbf{Q}^{-1}\mathbf{x} \geq 0. \end{aligned}$$

He then uses *a positive vector* $\boldsymbol{r}$ to hide $\mathbf{x}$ and obtains

$$\begin{aligned} & \min \mathbf{c}^T\mathbf{Q}(\mathbf{Q}^{-1}\mathbf{x} + \mathbf{r}), \\ \text{s.t.} \quad & \mathbf{M}_1\mathbf{Q}(\mathbf{Q}^{-1}\mathbf{x} + \mathbf{r}) = \mathbf{b}_1 + \mathbf{M}_1\mathbf{Q}\mathbf{r}, \\ & \mathbf{M}_2\mathbf{Q}(\mathbf{Q}^{-1}\mathbf{x} + \mathbf{r}) \leq \mathbf{b}_2 + \mathbf{M}_2\mathbf{Q}\mathbf{r}, \\ & (\mathbf{Q}^{-1}\mathbf{x} + \mathbf{r}) \geq \mathbf{r}. \end{aligned}$$

Setting $\mathbf{z} = \mathbf{Q}^{-1}\mathbf{x} + \mathbf{r}$ and taking a strictly positive diagonal matrix $\mathbf{S}$ (a diagonal matrix where the entries on the main diagonal are strictly positive), the client obtains

$$\begin{aligned} & \min \mathbf{c}^T\mathbf{Q}\mathbf{z}, \\ \text{s.t.} \quad & \mathbf{M}_1\mathbf{Q}\mathbf{z} = \mathbf{b}_1 + \mathbf{M}_1\mathbf{Q}\mathbf{r}, \\ & \mathbf{M}_2\mathbf{Q}\mathbf{z} \leq \mathbf{b}_2 + \mathbf{M}_2\mathbf{Q}\mathbf{r}, \\ & \mathbf{S}\mathbf{z} \geq \mathbf{S}\mathbf{r}, \\ & \mathbf{z} \geq 0 \text{ (see the above definitions of } \mathbf{Q} \text{ and } \mathbf{r}). \end{aligned}$$

Set $\mathbf{c}'^T = \mathbf{c}^T\mathbf{Q}$ and

$$\mathbf{M}' = \begin{pmatrix} \mathbf{M}_1\mathbf{Q} & 0 \\ \mathbf{M}_2\mathbf{Q} & \mathbf{A} \\ -\mathbf{S} & \end{pmatrix}, \ \mathbf{b}' = \begin{pmatrix} \mathbf{b}_1 + \mathbf{M}_1\mathbf{Q}\mathbf{r} \\ \mathbf{b}_2 + \mathbf{M}_2\mathbf{Q}\mathbf{r} \\ -\mathbf{S}\mathbf{r} \end{pmatrix}$$

where $\mathbf{A}$ is a permutation matrix representing slack-variables. Hence, the client can rewrite the program as follows:

$$\min \mathbf{c}_s'^T\mathbf{z}_s, \text{ s.t. } \mathbf{M}'\mathbf{z}_s = \mathbf{b}', \ \mathbf{z}_s \geq 0,$$

where $\mathbf{c}'_s$ is $\mathbf{c}'$ with added zeros for the slack-variables and $\mathbf{z}_s$ is the variable vector ($\mathbf{z}$ with added slack-variables). To hide the contents of $\mathbf{M}'$ and $\mathbf{b}'$, the client uses a nonsingular matrix $\mathbf{P}$ and with $\widehat{\mathbf{M}} = \mathbf{PM}'$ and $\widehat{\mathbf{b}} = \mathbf{Pb}'$ and obtains

$$\begin{aligned} & \min \mathbf{c}_s'^T\mathbf{z}_s, \\ \text{s.t.} \quad & \widehat{\mathbf{M}}\mathbf{z}_s = \widehat{\mathbf{b}}, \\ & \mathbf{z}_s \geq 0. \end{aligned}$$

Finally, the client outsources the above problem to the cloud server. As

$$\mathbf{z} = \mathbf{Q}^{-1}\mathbf{x} + \mathbf{r},$$

the resulting $\mathbf{x}$ can be obtained from $\mathbf{z}$ by calculating

$$\mathbf{x} = \mathbf{Q}(\mathbf{z} - \mathbf{r}).$$

Notice that in the Dreier-Kerschbaum scheme the nonnegativity constraints $\mathbf{z}_s \geq 0$ has explicitly specified. But it is a pity that the authors [11] did not pay more attentions to the specification although they cited the Dreier-Kerschbaum's work.

The designing art in the scheme can be depicted as follows

$$\begin{aligned} \mathbf{x} \quad & \xrightarrow{\text{affine transformation}} \quad \mathbf{z} = \mathbf{Q}^{-1}\mathbf{x} + \mathbf{r} \\ & \xrightarrow{\text{adding slack-variables}} \quad \mathbf{z}_s = (\mathbf{z}^T, z_{n+1}, \cdots, z_{n+k})^T. \end{aligned}$$

Clearly, the cloud server cannot recover $\mathbf{x}$ from $\mathbf{z}_s$ because $\mathbf{Q}, \mathbf{r}$ are the session keys randomly picked by the client.

5 Conclusion

We point out that the procedure for determining the leaving basic variable in the simplex method requires that all variables are subject to nonnegativity. One must draw a clear distinction between the functional inequality constraints and the nonnegativity constraints.

Notice that deriving the augmented form of a standard form for a linear programming problem is very easy. It can be solely done by the client himself even though who is assumed to be of weak computational capability.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Project 61303200, 61411146001). The authors gratefully acknowledge the reviewers for their valuable suggestions.

References

[1] Z. J. Cao and L. H. Liu, "Comment on 'harnessing the cloud for securely outsourcing large-scale systems of linear equations'," *IEEE Transactions on Parallel Distribution Systems*, vol. 27, no. 5, pp. 1551–1552, 2016.

[2] Z. J. Cao and L. H. Liu, "The paillier's cryptosystem and some variants revisited," *International Journal of Network Security*, vol. 19, no. 1, pp. 89–96, 2017.

[3] F. Chen, T. Xiang, and Y. Y. Yang, "Privacy-preserving and verifiable protocols for scientific computation outsourcing to the cloud," *Journal of Parallel Distribution Computing*, vol. 74, pp. 2141–2151, 2014.

[4] J. Dreier and F. Kerschbaum, "Practical privacy-preserving multiparty linear programming based on problem transformation," in *Proceedings of IEEE International Conference on Privacy, Security, Risk, and Trust*, pp. 916–924, Boston, MA, USA, Oct. 2011.

[5] F. Hillier and G. Lieberman, *Introduction to Operations Resarch (9th edition)*. USA: McGraw-Hill Higher Education, 2010.

[6] W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[7] X. Y. Lei, X. F. Liao, T. W. Huang, and F. Heriniaina, "Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud," *Information Sciences*, vol. 280, pp. 205–217, 2014.

[8] X. Y. Lei, X. F. Liao, T. W. Huang, H. Q. Li, and C. Q. Hu, "Outsourcing large matrix inversion computation to a public cloud," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 78–87, 2013.

[9] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing," *International Journal of Network Security*, vol. 18, no. 4, pp. 650–666, 2016.

[10] D. Marinescu, *Cloud Computing Theory and Practice*. USA: Elsevier, 2013.

[11] H. X. Nie, X. F. Chen, J. Li, J. Liu, and W. J. Lou, "Efficient and verifiable algorithm for secure outsourcing of large-scale linear programming," in *Proceedings of 28th IEEE International Conference on Advanced Information Networking and Applications (AINA'14)*, pp. 591–596, Victoria, BC, Canada, May 2014.

[12] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of Advances in Cryptology (EUROCRYPT'99)*, pp. 223–238, Prague, Czech Republic, May 1999.

[13] S. Salinas, C. Q. Luo, X. H. Chen, and P. Li, "Efficient secure outsourcing of large-scale linear systems of equations," in *Proceedings of 2015 IEEE Conference on Computer Communications (INFOCOM'15)*, pp. 1035–1043, Hong Kong, China, Apr. 2015.

[14] J. Singh, "Cyber-attacks in cloud computing: A case study," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 78–87, 2014.

[15] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *Proceedings of 2011 IEEE Conference on Computer Communications (INFOCOM'11)*, pp. 820–828, Shanghai, China, Apr. 2011.

[16] C. Wang, K. Ren, J. Wang, and Q. Wang, "Harnessing the cloud for securely outsourcing large-scale systems of linear equations," *IEEE Transactions on Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1172–1181, 2013.

[17] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[18] M. Zareapoor, P. Shamsolmoali, and M. A. Alam, "Establishing safe cloud: Ensuring data security and performance evaluation," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 88–99, 2014.

Zhengjun Cao is an associate professor of Department of Mathematics at Shanghai University. He received his Ph.D. degree in applied mathematics from Academy of Mathematics and Systems Science, Chinese Academy of Sciences. He served as a post-doctor in Computer Sciences Department, Université Libre de Bruxelles, from 2008 to 2010. His research interests include cryptography, discrete logarithms and quantum computation.

Lihua Liu is an associate professor of Department of Mathematics at Shanghai Maritime University. She received her Ph.D. degree in applied mathematics from Shanghai Jiao Tong University. Her research interests include combinatorics, cryptography and information 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.