IJNS
International Journal of Network Security

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

Vol. 23, No. 4 (July 2021)

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

Volume: 23, No: 4 (July 1, 2021)

International Journal of Network Security

Research on E-book Security Tracking Schemes

Cheng-Ying Lin[1,2], Chia-Chun Wu[3], and Min-Shiang Hwang[1,4]
(Corresponding author: Min-Shiang Hwang)
Department of Computer Science & Information Engineering, Asia University[1]
The Ph.D. Program in Artificial Intelligence, Asia University[2]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taichung, Taiwan, ROC
Department of Industrial Engineering and Management, National Quemoy University[3]
No. 1, University Rd., Jinning Township, Kinmen County 892, Taiwan, ROC)
Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan[4]
Email: mshwang@asia.edu.tw
(Invited Paper; First Online June 22, 2021)

Abstract

There are many business models for e-books, such as billing by time, billing by the hour, and database authorization. Copyright authorization and pricing methods are also different. How to assist the industry in formulating appropriate authorization and pricing mechanisms under different operating models is the key to the overall development of e-book services in the future. This research will be divided into two parts. The first one will start with the E-book authorization tracking mechanism. The purpose of this research is to solve the current problem of unauthorized and illegal distribution of e-books. The content consists of three parts. The first is privacy protection technology for legitimate users. Even sellers cannot obtain the personal information of legitimate users. The second part is the tracking mechanism for illegally authorized users. When the user performs illegal authorization, the e-book provider can use the tracking mechanism to find the illegal person. The third part is to build an e-book authorization management mechanism for different business models to achieve authorization tracking. The second topic will research the "e-book pricing tracking mechanism." The purpose is to design appropriate securities pricing mechanisms for different operating models. The project's content is divided into three sub-topics: E-book pricing and tracking mechanism, E-book time-limited pricing management mechanism and building an E-book payment transaction server. The research will understand users' browsing habits and compare them with industry interests. To develop a win-win pricing system, we will formulate an appropriate charging pricing system for different browsing methods and adopt the system according to the situation.

Keywords: Authorized Tracking; Business Models; Charging Mechanism; E-Book

1 Introduction

In recent years, digital multimedia technology has become increasingly mature due to the development of science and technology. Many traditional books have been presented in digital form. E-books are different from traditional books in the past in that they can provide multimedia content and even interactive multimedia [20]. Nowadays, we can integrate multimedia resources such as text, images, and sound to present information, and e-books can integrate multimedia resources. Many traditional books are also presented digitally, and readers can also enjoy a lot of reading pleasure. However, the development of e-books is currently restricted. In addition to copyright protection, illegal copying of e-books and management of transaction tracking methods are also critical issues. Rampant piracy and illegal authorized use make the copyright protection of e-book content face considerable challenges.

At present, most publishers are engaged in the publishing of traditional books, and it is still difficult to enter the e-book industry. First of all, It must outsource the production of e-books, and the format of e-books is also essential. If there is no standard, circulation and production costs will increase—one of the more popular e-book standards, such as Epub. With the universal standards for e-books, it can develop e-book servers channels in the future, and publishers can also have their own unique business models. This research aims to study the business model of e-books, the authorization tracking mechanism of transactions, and the multiple pricing mechanisms of e-books.

The research will combine the relevant technologies of other topics, including e-book content integrity protection technology, e-book image security protection mechanism, e-book 3D model protection mechanism, and e-book cloud hierarchical access technology. This makes it more convenient and safer for publishers to manage digital rights

on the server. In addition, this research has promoted the specific development of the digital publishing industry and promoted cross-industry cooperation between digital publishing content and information and communication services.

The biggest problem facing publishers is that consumers copy e-books at will and illegally authorize them. After consumers purchase e-books, it is easy to copy e-books and send them to others for viewing, which causes authorization management problems. To solve this problem, current e-book operators will perform identity authentication when consumers purchase e-books and use the real identity of users as an important basis for future e-book authorization tracking. In other words, the current pricing and authorization mechanism for e-books is based on the current consumer's identity as verification, and there is no anonymity protection.

Countries worldwide are paying more and more attention to personal privacy, and consumers naturally do not want e-book suppliers or other consumers to know their reading records. Therefore, anonymous pricing and authorization mechanisms for e-books are indeed necessary—however, the issue of illegal consumer authorization conflicts with the need for anonymity. Therefore, taking into account the tracking of illegal authorized persons by e-book manufacturers and reducing consumers' concerns about privacy leakage has become a critical issue. This research has developed a set of pricing and authorization tracking technology on this operating server to develop and use appropriate business models for various e-book operating companies.

The first topic of this research is to study the e-book authorization tracking mechanism. On the premise of legal use by consumers, our mechanism can guarantee consumers' privacy. In other words, we can provide consumers with anonymous demand. But once we discover that consumers may illegally authorize or illegally copy e-book files, we can track them through the established mechanism to find out who is the original illegal spreader of e-book files.

In addition, the pricing method of e-books is also a critical issue. There are many business models and pricing for e-books, such as billing or hourly charges. In the pricing process, we must also provide consumers with the necessary privacy protection. In addition to the development of multimedia resources, the content of e-books has become more and more diversified, and many value-added services of e-books have emerged. To develop a win-win pricing system, protect the rights of the industry and consumers, and target different browsing methods. We will propose an appropriate fee pricing scheme. It is precise because e-books can integrate multimedia resources, and it is also possible to embed the tracking verification information developed in this research into e-books.

Based on the above research goals, we will conduct two research topics. The first topic is the e-book authorization tracking mechanism. This research topic consists of three parts—first, design user privacy protection technology based on the existing e-book business model. Thus, even sellers cannot obtain the personal information of legitimate users. The second part is to adopt a tracking mechanism for illegally authorized users to protect users' privacy under normal and legal behaviors. But when committing illegal acts, e-book providers can use the tracking mechanism to find out the identity information of illegal users. The third part is to build an e-book authorization management mechanism. Finally, establish an e-book business model to achieve e-book authorization tracking of different business models. The second topic, research on the e-book business model and its pricing tracking mechanism. It is divided into three sub-topics: E-book pricing management mechanism limited by frequency, E-book pricing management mechanism limited by the time, and establishment of e-book payment transaction server. This topic understands users' browsing habits, compares them with industry interests, and develops a win-win pricing system. Then, develop corresponding charging systems for different browsing methods, and adopt the system according to the situation.

This paper is organized as follows. Section 2 introduces the related works on blind signatures, illegals use of tracking schemes, and user authentication schemes based on frequency and time limit. In Sections 3 and 4, we will propose two research issues for E-book authentication tracking mechanism and E-book pricing tracking mechanism. Finally, a conclusion is conducted in Section 5.

2 Related Works

The following will discuss the documents related to the blind signature and illegal use tracking mechanism and the user verification mechanism based on the number and time limit to understand the current development trend and practice.

2.1 Blind Signatures and Illegal Use of Tracking Schemes

In a good e-commerce environment, the merchants cannot know which products each customer has purchased to protect consumer privacy. In other words, when making a payment, a merchant only needs to verify whether the electronic money paid by the consumer is correct and valid. It is impossible to know who the holder of electronic money is. However, once consumers illegally use electronic money, such as reusing electronic money, the system must have a mechanism to track the users of such illegal electronic money.

However, electronic money mechanisms that can track users often make users' privacy disappear. For example, E-book providers can track who pays through a single digital message on each e-currency. In this way, the merchant can know exactly what the consumer bought. This also allows users with traceable payment mechanisms to no longer have privacy. Therefore, many scholars men-

tioned in the past that the protection of consumer privacy must be anonymous, data privacy, and identity identification. Brands [1] uses the blind signature method proposed by Chaum in 1983 [3] to implement untraceable electronic money technology for paying users. The purpose of this method is that the signer and other verifiers can only verify whether the digital signature of a certain document is correct but cannot find out any relationship between the document and the document when signing. This technology also satisfies the untraceability of electronic money [4]. This feature can protect the privacy and security of users and can detect repeated operations.

Chaum's blind signature technology is based on RSA's public cryptographic system [3]. The following describes the signature method. There are two participating entities under this framework: User and Singer. The steps are listed as follows:

1) First, the user randomly selects a random value r and saves it properly.

2) Next, the user uses the signer's public key (e, n) and the selected random value r to blind the message m to be signed, thereby making $M = mr^s \bmod n$. Then the blind message M is sent to the signer.

3) After receiving the blind message M, the signer digitally signs M with its private key d, and the digital signature S_M obtained is $S_M = M^d \bmod n$. The signer then sends this blind signature back to the user.

4) Because S_M meets the following conditions:

$$\begin{aligned} S_M &= M^d \bmod n \\ &= (mr^e)^d \bmod n \\ &= (m^d r^{ed}) \bmod n \\ &= (m^d r)^d \bmod n. \end{aligned}$$

Therefore, after receiving the S_M, the user can easily use the random number r of his/her own choice to open the blind. Obtain the digital signature S_m of the message m, as follows:

$$\begin{aligned} S_m &= S_M r^{-1} \bmod n \\ &= (m^d r) r^{-1} \bmod n \\ &= m^d \bmod n. \end{aligned}$$

Hwang *et al.* proposed an untraceable blind signature scheme based on the RSA cryptosystem [6]. They applied the extended Euclidean algorithm to the blind signature scheme. The security of their proposed scheme is the same as that of the RSA cryptosystem, depending on the difficulty of solving the factorization problem. Lee *et al.* proposed a new blind signature based on the discrete logarithm problem [12]. Yin *et al.* proposed a blind signature scheme based on the identity of elliptic curves [21]. They applied dot product operations on elliptic curves instead of bilinear pair operations, which reduces computational overhead. In the past two decades, many blind signatures have been proposed [7, 8, 14].

Using the above steps, the user can successfully obtain the digital signature of the message from the signer. The signer and other verifiers can verify the correctness of this signature. However, the signer cannot know which user this message m belongs to. This is because the message m during the verification period is different from the message M signed by the signer ($M \neq m$). Therefore, the signer cannot track the source of this message. The method of signing achieves the purpose of protecting user privacy and satisfies the security requirements of message authentication and integrity.

Subsequently, Brands proposed a double-spending preconstraint mechanism, as long as the consumer will expose repeated payments [1]. There are three entities in this structure: user (consumer), store, and bank. First, it must establish the system, and then the user must establish an account in the bank.

Saputra *et al.* proposed a general model of electronic cash [17]. The development of this model uses logical modeling based on Inenaga *et al.* [11]. The model consists of three sub-models: system model, process model, and attribute model. Their model can be used as a basis for security assessment and covers a wider range of electronic cash scheme changes than the model of Inenaga et al.

2.2 User Authentication Schemes Based on Frequency and Time Limit

In an e-commerce environment, users must verify with the system to avoid privacy and security issues before using the service to prove that they are legitimate users. In many e-commerce applications, password authentication methods are widely used because of their simplicity and ease of use. In addition to improving the security of the system, technical research on user authentication will also benefit the development of e-commerce. In business applications, one of the key issues is how to identify and verify user objects. In addition, how to charge users is another important issue. To protect the interests of the industry and the quality of service, the industry can allow users to access resources through a time limit or frequency limit mechanism. Therefore, the user must first pay the service fee. After the verification is passed, the user can obtain the resources to be used to protect the security mechanism.

If the operator needs to control user access to resources, he must re-maintain and record the usage of each user. If the usage limit reaches the limit, the user needs to pay again to pass the verification. Therefore, based on the above situation, authentication and restricted use will increase the industry's costs to maintain user data. Obviously, this method cannot be effectively applied to the situation of e-commerce.

For e-books to be more widely used in e-commerce, a safe and effective tracking mechanism needs to be de-

signed. In addition to verifying the user's identity, Santis *et al.* also suggest that it should use time limits to control the use of legitimate users [16]. In addition, e-books also need to limit the user's use time. Due to the current prevalence of mobile vehicles, e-book applications have also emerged. Due to the limitation of the computing power of mobile in-vehicle devices, this research will develop mechanisms and methods to minimize computing resources.

Previous researches about the general authentication system usually need to maintain effective identity authentication and password forms. Only after passing the identity verification and password verification, the user can use the service. However, many studies point out that this method is quite dangerous. The intruder can obtain the password in other ways to achieve the purpose of intrusion. Many methods have replaced the maintenance form and replaced it with a verification form to avoid this password leakage problem. The password field is encrypted by a one-way hash function or another encryption algorithm in this verification form. There may also be another problem. The user and the system are not secure enough. Attackers can intercept past login information to impersonate legitimate users. Therefore, many studies are still trying to solve such problems, but they require more computing and network traffic. In other words, past research still cannot be effectively applied to the behavior of e-commerce. Therefore, to more effectively promote the commercial application of e-books, this research proposes to improve the above shortcomings and achieve the following advantages:

1) The system no longer needs to store and maintain login forms.
2) The login verification process can prevent replay attacks.
3) The system is easy to use.

In such an environment, to protect the rights and interests of copyright owners, only legitimate users can browse the contents of e-books. And to ensure that the file data will not leak out. After the verification expires, how to ensure that the user cannot use the deleted files again? Tezuka *et al.* proposed a time-based and strategy-based scheme [19]. Perlman proposes a time-based approach: when a file is created, the user will access it within a limited time [15]. When the valid time expires, the user can no longer calculate the encryption key. This means that the user will no longer be able to access the file. Tang *et al.* explain that the policy-based method is: when a file is created due to a policy, the system will generate a pair of public and private keys [18]. When an employee leaves the team, the system will delete the public and private keys used. Under this policy, the user can no longer access the file. But every time you change the policy, you need to delete and regenerate the key. In other words, the inconvenience of key storage and the amount of calculation will be a critical issue.

In addition, in previous studies, we have also proposed a secure authentication transaction mechanism for e-commerce and proposed related technologies. The basic concept is as follows. First, the customer must register as a valid user with the service provider, and the service provider will give valid authentication information. When a user wants to obtain a service from a service provider, he must provide relevant information to verify the legality of the use. In the past, this mechanism was successfully applied to online games. In addition to verifying user identities, the online game company's system also charges fees based on game content. Users must prepay to get used tickets. After the authentication is passed, the user can obtain the authorization to play the game. The system will verify and record the number of games. When the number of times reaches the limit, if the user wants to continue the game, he needs to pay another fee to get the ticket.

3 Topic 1: Research and Development of E-book Authorization Tracking Mechanism

There are three sub-topics in this research.

1) Research and development of privacy protection mechanisms for e-book users.
   The purpose of this research topic is to protect the privacy of legitimate users by contacting sellers who cannot know which legitimate users purchased this book, thereby protecting the privacy of legitimate users.

   This mechanism has two subjects: the user and the e-book provider. The method proposed by Brands carries out user privacy protection. The research methods and steps of this research are as follows (as shown in Figure 1):

   a. The user is registered as a legal user at the e-book supplier.
   b. Legally apply for e-book authorization.
   c. Determine whether it is a user who has successfully registered.
   d. If the authorization is successful, but the e-book content is illegally reprinted or shared, it will enter the illegal user tracking mechanism.

   The above is the first part of the research topic. We refer to various user privacy protection mechanisms and the application of blind signatures in privacy protection from many documents and formulate regulations that can resist various attacks and meet various security requirements. The secure e-book value-added service mechanism is expected to solve user privacy data being tracked in the past and protect user privacy better than previous mechanisms.

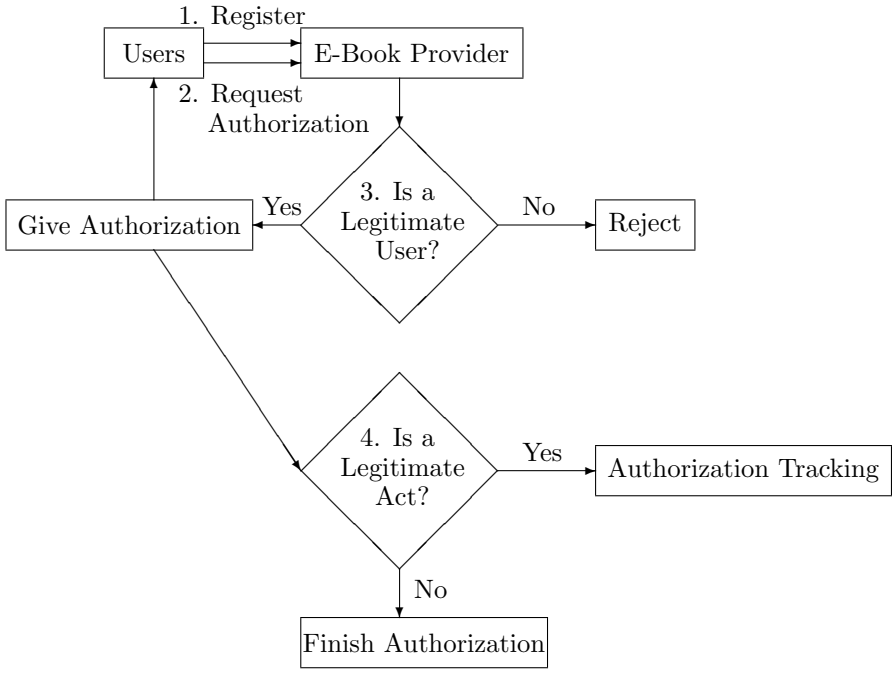


Figure 1: E-book protection user privacy flowchart

2) Research and development of tracing illegal users with a tracking mechanism.
There are many mechanisms for protecting the use of digital content. The encryption mechanism is used to confirm whether users use digital content legally and protect the digital content, that is, the copyright of e-books, through watermarking, limiting the time of use, or through legal authorization. But sometimes, it is impossible to connect to the Internet anytime and anywhere to obtain legal authorization. Therefore, one of the research topics is to protect copyright through the encryption of digital content. This issue aims to activate the authorization tracking mechanism when users illegally reprint or exchange authorized e-books. This topic can track illegal users but will not reveal the privacy of legitimate users.

According to the framework proposed by Brands [16], when a user makes a withdrawal, the bank will send electronic money to the user. Therefore, the research topic applies the Brands architecture to the e-book environment [16]. If users repost or share, they will start up the tracking mechanism. The e-book supplier can obtain $g_1^{(r_1-r_1')/(r_2-r_2')}$ from two re-used e-books through calculation and prove that $(r_1-r_1')/(r_2-r_2')$ mod q is a user authorized for dual-use or dual-sharing e-books only once.

To realize the untraceability of user privacy, the e-book provider of a legitimate user cannot know any user information. For illegal users, the project proposes a method that can use post-confirmation to track only illegal users. This proposed method can take the system offline and online. Once an illegal user uses an e-book, immediately detect whether there is an illegal user, the steps are as follows (as shown in Figure 2):

a. Illegal users must first register with the e-book supplier to obtain an account;
b. E-book authorization request;
c. The e-book supplier will first authorize;
d. Once illegal users reprint and share;
e. The e-book supplier will detect illegal users;
f. Access is denied.

3) Design of e-book authorization management server.
This research topic is the construction of an e-book authorization management server, which aims to provide services for the authorization, protection, and verification of e-book copyright.

a. Construction of verification database.
In addition to providing authorization services for e-books and protecting user privacy, the establishment of the server also needs to open a tracking mechanism when e-books are used illegally (see Figure 3). If users use this server, they must first register their identity and become a member after registration. When a user obtains an e-book on this server, the server will store various member information in the information system database, including user anonymous ID, authorized e-book serial number, authorization code, and timestamp. The purpose of recording is: (1) If later use by illegal readers can be

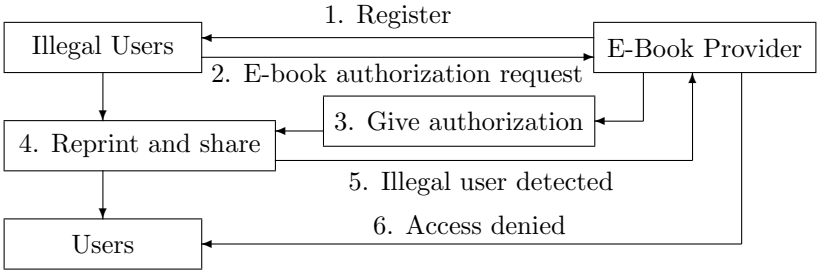


Figure 2: Authorization tracking mechanism

found immediately. (2) If readers find that the authorized e-book no longer exists, readers can also check on this server to prove that the readers have been authorized to obtain the qualification. (3) There are records in the database, which can be interactively queried with the e-book pricing tracking mechanism.

b. Establish an environment for authorization tracking mechanism.
Save these four kinds of information: anonymous ID, authorized e-book (serial number, e-book file, or hyperlink URL), authorization code, and timestamp.

c. Construction of e-book verification process.
This research topic uses the four types of data recorded above to establish a complete e-book verification process based on its previously proposed user privacy protection mechanism.

4 Topic 2: Research on E-book Pricing Tracking Mechanism

The second research topic is expected to achieve the following research goals:

1) The e-book pricing management mechanism is limited by the number of times.

2) Time-limited e-book pricing management mechanism.

3) Establishment and evaluation test of e-book secure payment transaction.

When e-book publishers want to publish copyrighted content in the system architecture, they will obtain relevant information required for user verification through the managed device. The e-book trading server will verify whether the e-book user is a legally authorized user based on the received information. If the user is authenticated as a legitimate user, the user can read the e-book normally; On the contrary, if the user is authenticated as an illegal user, including users whose authorization has expired, they will be denied access to e-book content until the user re-authorizes.

There are three main entities in the proposed e-book authorization management scheme. As shown in Figure 13.

1) E-book authorization management server: Place e-books with various contents through the cloud space to provide user authorization.

2) Verification Server: Responsible for verifying whether it is a legally authorized user.

3) User: It is a restricted device, and the user can browse the contents of the e-book after authentication.

We proposed an e-book authorization management mechanism with a limited number of times and time. Frequency limitation means that users can only use it a certain number of times. The time limit is that readers can access e-books normally within a certain time interval. In other words, when there is an expired request, you will not be able to obtain any e-book access file permissions. Based on this architecture, we can establish an e-book business model to make an e-book authorization management server more reliable.

As for how to add frequency and time factors to key management, we use the following three mathematical theories to achieve:

1) Quadratic residual: Assuming that y and p are integers and the greatest common factor is 1, if $x^2 \equiv y \bmod p$ has an integer solution, then y is called the quadratic residual of mod$p$ (QR); if it is not true, then y is not the QR of mod$p$ (NQR) [22]. For example,

$$\begin{aligned} 1^2 &\equiv 6^2 \equiv 1 \bmod 7, \\ 2^2 &\equiv 5^2 \equiv 4 \bmod 7, \\ 3^2 &\equiv 4^2 \equiv 2 \bmod 7. \end{aligned}$$

From the above equations, we can see that 1, 2, and 4 are the Quadratic residual of mod7. However, 3, 5, and 6 are non-Quadratic residual of mod7.

2) One-way hash function: The one-way function method is usually used to verify the integrity of the message. When there is a function $y = H(x)$, if x is given, we can quickly calculate y. But if we only know y, it is difficult to derive x [2,10].

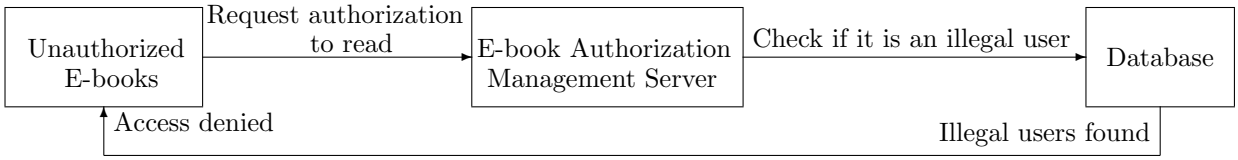


Figure 3: Illegal users can be found immediately

3) Message authentication code: The message authentication code is used to verify the source and integrity of the message. When the user with the key receives the message, he/she can know whether the message is correct [5].

In the research topic, we proposed a secure and efficient authentication mechanism for e-books as follows: First of all, users must first pay the operator to become legitimate users. The operator will provide valid identification (ID) and password (PW), as well as the use of tickets (TK). When a user requests a service from a provider, the user needs to log in his/her ID, password, and TK and provide the provider to verify its legitimacy.

To meet the security requirements in the application, there is no need to maintain any verification tables on the system [9, 13], and each TK will have a limit on the number of times and limit times of use.

According to the above design concept, we use quadratic residual to achieve the research goals. Therefore, our proposed method is designed based on the difficulty of solving the modulo square root. Specifically, the proposed scheme can be divided into three phases: Registration phase, login phase, and authentication phase.

In addition, the system will set the initial values of the system parameters before execution. First, the provider (SP) will choose two very different odd prime numbers, p and q, which need to satisfy $p \equiv q \equiv 3 \bmod 4$, and finally, the provider will calculate $n = pq$. Next, the provider discloses n, but p and q should be kept secure.

Next, we will describe the details of each phase.

Registration Phase.

First, in the registration phase, users need to register with the e-book provider (SP). Then, the user can obtain a ticket for future use. Tickets have a certain number and time limit. Readers need to perform the following steps for each newly applied user to obtain a ticket from the SP.

1) Each user Ui requests service from SP and pays the fee. The service request message includes personal information, service fees, and the number of tickets ordered by the user.

2) The SP will generate a unique user identity (ID_i) and an integer (r_i) for each user (U_i). r_i needs to meet:

$$\begin{aligned} r_i^{\frac{p-1}{2}} &\neq 1 \bmod p, \\ r_i^{\frac{q-1}{2}} &\neq 1 \bmod q. \end{aligned}$$

Here, $r_i \in Z_n$, $\gcd(r_i, n) = 1$.

3) Let t be the number of tickets ordered by the user (U_i). SP needs to calculate

$$\begin{aligned} PW_i^j &\equiv r_i^{2^j} \equiv (r_i^{2^{j-1}})^2 \bmod n, \\ \alpha_j &= MAC_p(ID_i||TID_j||ED_j||PW_i^{(j-1)}). \end{aligned}$$

Where $j = 1, 2, \cdots, t$. The $||$ symbol indicates the connection of two integers. TID_j is the unique serial number in the ticket TK_j. ED_j is the expiration information of TK_j. Each TK_j contains $\{TID_j, ED_j, \alpha_j\}$.

4) Finally, the SP will send $ID_i, TK_1, TK_2, \cdots, TK_t$, and PW_i^t through the secure channel. Until the reader receives this information, the U_i will be stored on the reader device.

Login Phase.

During the login phase, the proposed scheme allows readers to use the ordered tickets for a limited time freely. Each ticket contains a password, which readers can use to prove the legitimacy of the reader. If the ticket expires, the reader needs to obtain another unused ticket to verify its legitimacy. After the reader's tickets are used up, he/she needs to purchase new tickets at the SP to obtain the right to use them.

At this phase, it is assumed that the reader U_i has obtained a valid ticket TK_α and contains the $PW_i^{(a)}$ (Here, $a \in [1, t]$), and at a certain time, TS has obtained the service from the SP. U_i will perform the following steps to confirm that he/she is indeed a legitimate user:

1) Time series TS was taken by U_i.
2) U_i calculates $\beta_a = MAC_{\alpha_a}(ID_i||TS)$.
3) U_i sends ID_i, $PW_i^{(a)}$, TID_α, ED_α, TS, and β_α to SP for future verification.

Authentication Phase.

In the authentication phase, assuming that the SP

receives the ID_i, $PW_i^{(a)}$, TID_α, ED_α, TS, and β_α from the U_i in timestamp TS', then the SP will do the following steps to verify the legitimacy of the user's login request.

1) The SP will first check the validity of the received timestamp and set ΔTS as the delay time between SP and used transmission. When $(TS' - TS) > \Delta TS$, hackers may steal the login request, and the SP will reject the service request.
2) The SP checks the format of the received ID and TID_α. If the format is incorrect, the SP will reject the service request.
3) The SP will check whether the ticket has been used. If the currently verified ticket TID_α is already in use, the SP will reject the service request.
4) The SP will check whether the date of the ticket has expired. If $TS > ED_\alpha$ indicates that the ticket has expired, the SP will reject the service request.
5) If all the above verifications are passed, SP will calculate the 4 possible square roots of $PW_i^{(a)}$: $R_\alpha^{(1)}$, $R_\alpha^{(2)}$, $R_\alpha^{(3)}$, and $R_\alpha^{(4)}$ as follows:

$$\begin{aligned} R_\alpha^{(1)} &\equiv uqq^* + vpp^* \bmod n, \\ R_\alpha^{(2)} &\equiv uqq^* - vpp^* \bmod n, \\ R_\alpha^{(3)} &\equiv -uqq^* + vpp^* \bmod n, \\ R_\alpha^{(4)} &\equiv -uqq^* - vpp^* \bmod n. \end{aligned}$$

6) Next, SP will calculate 4 α_a:

$$\alpha_a^{(j)} = MAC(ID_i||TID_a||ED_a||R_a^{(j)}),$$

where $j = 1, 2, 3, 4$.

7) The SP decides which candidate solution satisfies the $MAC_{\alpha_a^{(j)}}(ID_i||TS) = \beta_a$.
8) If all service requests pass the above verification, the SP will disclose the TID_a and make a $PW_i^{(a-1)} = R_a$. Then SP will send $PW_i^{(a-1)}$ to U_i.

5 Conclusions

Due to the copyright and purchaser authorization issues of e-books, this part has its own particularities, and related researches are rare in the past. This research proposes the authorization management and charging pricing mechanism of e-books. It can protect consumers' privacy and track users who are illegally authorized to use. And developed a set of multiple counting and time-limited e-book pricing mechanisms. The first research topic is based on blind signature technology. Under normal use conditions, it can protect the user's (reader) privacy. However, in the case of illegal use, it may expose the identity of the illegal user again. This technology is quite innovative in theory, and it can also develop a tracking mechanism for illegal authorization. The second research topic is the research and development of e-book pricing methods and their operating business models. We have developed a pricing mechanism with frequency limits and time limits. Through the design of the authentication mechanism, users can efficiently access e-books with privacy protection.

This research can be applied to the authorization and pricing of e-books and solve the problems of e-books in authorization management and pricing tracking. It provides a possible development direction for the business operation model of the e-book industry. It can also be extended to related applications of e-commerce or digital rights management in the future and contribute to industrial upgrading. It can also apply the research to various special applications, such as publishing companies, record companies, or multimedia websites. In the future, authorization management servers and pricing tracking servers can also be applied in related fields. The field of digital rights management.

Acknowledgments

The Ministry of Science and Technology partially supported this research, Taiwan (ROC), under contract no.: MOST 109-2221-E-468-011-MY3, MOST 108-2410-H-468-023, and MOST 108-2622-8-468-001-TM1.

References

[1] S. Brands, "Untraceable off-line cash in wallet with observers," *Lecture Notes in Computer Science*, vol. 773, pp. 302–318, 1994.
[2] T. Y. Chang, M. S. Hwang, and W. P. Yang, "A new multi-stage secret sharing scheme using one-way function", *ACM Operating Systems Review*, vol. 39, no. 1, pp. 48–55, Jan. 2005.
[3] D. Chaum, "Blind signatures for untraceable payments," in *Proceedings of Advances in Cryptology (CRYPTO'82)*, pp. 199–203, California, USA, 1982.
[4] D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," in *Proceedings of Advances in Cryptology (CRYPTO'88)*, pp. 319–327, California, USA, 1988.
[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] M. S. Hwang, C. C. Lee, Y. C. Lai, "An untraceable blind signature scheme", *IEICE Transactions on Foundations*, vol. E86-A, no. 7, pp. 1902–1906, July 2003.

[7] M. S. Hwang, C. C. Lee, Yan-Chi Lai, "Traceability on low-computation partially blind signatures for electronic cash", *IEICE Fundamentals on Electronics, Communications and Computer Sciences*, vol. E85-A, no. 5, pp. 1181–1182, May 2002.

[8] M. S. Hwang, C. C. Lee, Y. C. Lai, "Traceability on Stadler et al.'s fair blind signature scheme", *IEICE Transactions on Fundamentals on Electronics, Communications and Computer Sciences*, vol. E86-A, no. 2, pp. 513-514, 2003.

[9] M. S. Hwang, J. W. Lo, S. C. Lin, "An efficient user identification scheme based on ID-based cryptosystem", *Computer Standards & Interfaces*, vol. 26, no. 6, pp. 565–569, 2004.

[10] M. S. Hwang and P. C. Sung, "A study of micropayment based on one-way hash chain", *International Journal of Network Security*, vol. 2, no. 2, pp. 81–90, Mar. 2006.

[11] S. Inenaga, K. Oyama, and H. Yasuura, "Towards modeling stored-value electronic money systems," *Information and Media Technologies*, vol. 6, no. 1, pp. 25–34, 2011.

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

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

[14] C. C. Lee, W. P. Yang, M. S. Hwang, "Untraceable blind signature schemes based on discrete logarithm problem", *Fundamenta Informaticae*, vol. 55, no. 3-4, pp. 307-320, 2003.

[15] R. Perlman, "File system design with assured delete," in *Third IEEE International Security in Storage Workshop (SISW'05)*, 2005.

[16] A. D. Santis, A. L. Ferrara, B. Masucci, "Enforcing the security of a time-bound hierarchical key assignment scheme," *Information Sciences*, vol. 176, no. 12, pp. 1684–1694, 2006.

[17] D. E. Saputra, S. Sutikno, and S. H. Supangkat, "General model for secure electronic cash scheme," *International Journal of Network Security*, vol. 21, no. 3, pp. 501–510, 2019.

[18] Y. Tang, P. C. Lee, C. S. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 903-916, 2012.

[19] S. Tezuka, R. Uda, and K. Okada, "ADEC: Assured deletion and verifiable version control for cloud storage," in *26th IEEE International Conference on Advanced Information Networking and Applications*, pp. 23-30, 2012.

[20] C. Y. Tsai, C. Y. Yang, I. C. Lin, and M. S. Hwang, "A survey of E-book digital right management", *International Journal of Network Security*, vol. 20, no. 5, pp. 998-1004, 2018.

[21] S. L. Yin, H. Li, S. Karim, and Y. Sun, "ECID: Elliptic curve identity-based blind signature scheme", *International Journal of Network Security*, vol. 23, no. 1, pp. 9-13, 2021.

[22] Y. Zhao, Q. Yang, B. Yang, "Provably secure partially blind signature scheme based on quadratic residue", *International Journal of Network Security*, vol. 17, no. 2, pp. 217-223, 2015.

Biography

Cheng-Ying Lin received the M.A. in Hochschule für Musik und Theater "Felix Mendelssohn Bartholdy" Leipzig, Germany, in 2015, and B.A. in National Hsinchu University of Education,Taiwan, in 2011. He is currently pursuing a Ph.D. degree in the Ph.D. Program in Artificial Intelligence, at Asia University, Taiwan. He was the owner and trombonist of the Brass Men ensemble. He was also served as a teacher and Wind Orchestra Conductor in many schools. His current research interests include Artificial intelligence, Computer music.

Chia-Chun Wu received a Ph.D. degree from the Department of Computer Science and Engineering, National Chung-Hsing University, Taichung, Taiwan, in 2011. He is currently an associate professor at the Department of Industrial Engineering and Management, National Quemoy University, Kinmen County, Taiwan. His current research interests include internet of things (IoT), database security, secret image sharing, mobile applications development, and digital image techniques.

Min-Shiang Hwang received M.S. in industrial engineering from National Tsing Hua University, Taiwan in 1988, and a Ph.D. degree in computer and information science from National Chiao Tung University, Taiwan, in 1995. He was a distinguished professor and Chairman of the Department of Management Information Systems, NCHU, during 2003-2011. He obtained 1997, 1998, 1999, 2000, and 2001 Excellent Research Awards from the National Science Council (Taiwan). Dr. Hwang was a dean of the College of Computer Science, Asia University (AU), Taichung, Taiwan. He is currently a chair professor with the Department of Computer Science and Information Engineering, AU. His current research interests include information security, electronic commerce, database, data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.

A Practical and Efficient Two-Part Edwards Curve Digital Signature for Mobile Networks

Mingrui Zhang[1,2], Bo Yang[1,2], Hongxia Hou[1,2,3], Meijuan Huang[1,2,4], and Yanwei Zhou[1,2]
(Corresponding author: Bo Yang)

School of Computer Science, Shaanxi Normal University[1]
Xi'an 710119, China
State Key Laboratory of Information Security (Institute of Information Engineering), Chinese Academy of Sciences[2]
Beijing 100093, China
School of Cyberspace Security, Xi'an University of Posts & Telecommunications[3]
Xi'an, Shaanxi 710121, China
School of Mathematics and Information Science, Baoji University of Arts and Sciences[4]
Baoji 721013, China
Email: byang@snnu.edu.cn
(Received Dec. 2, 2019; Revised and Accepted May 15, 2020; First Online May 26, 2021)

Abstract

In the IoT era, people can control their homes with a smartphone easily. To ensure the security of data transmission, digital signature protocols have been applied to the authentication of identity and messages. However, in the traditional method, a user's private key is directly stored on a mobile phone. So that the private key may be disclosed under various malicious attacks. A two-part signature on the Edwards curve is proposed in this paper to improve the security of the private key. A valid signature can be generated without reestablishing the whole private key. The security analysis of the protocol with standard assumptions is also presented. We implement this new protocol on PC and Android smartphones. The evaluation results demonstrate that our protocol runs faster in the key generation phase and has a smaller signature length. Therefore, this scheme can be effectively deployed in practice to protect the private key.

Keywords: EdDSA; Mobile Devices; Mobile Network; Signature; Smart Home

1 Introduction

Due to the development of mobile network and IoT (Internet of Things), living environment of people has been greatly improved. However, owing to constant and fast-paced cyber-attack evolution, The security of mobile network and smart mobile device are raising important. According to [2], in 2020, cyber-attacks will have launched in such a way that: Malicious attacks up to 1001 million, the web attacks 2746 million, and virus and malicious attacks 1585. The business loss caused by network attack will exceed $ 2000 millions. These challenges threaten Iot system security. For example, as a part of Internet of Things, smart home-smartphone system also suffers these security threats. In Smart home-Smartphone system, people can control their smart homes with smartApps on a smartphone through wireless network [4]. All kinds of Smartapps provided by various manufacturers have been able to connect and control smart home conveniently and quickly [10]. As users download and install smart apps from the mobile network, mobile network communication risks may threaten the security of the system.

In this smart home-smartphone system, smart home only executes instructions send by smart phone. In order to prove that instructions really comes from the real user rather than adversary, an effective method is to use authentication technology,such as digital signature. As shown Figure 1. Recently,some new authentication schemes have been proposed [15, 24]. Using smartphones to implement some existing authentication protocols is an effective solution. However, a user's private key always stored in mobile phone directly. There is a potential risk that smartphones could be attacked directly by malicious applications and user's private key may be disclosed. Or, the adversary may obtain the private key by analyzing the memory information of the mobile operating system [1, 17]. If the adversary obtains the key, then the adversary obtains control of the everything of the house.

In order to reduce the risk of the secret key to be stolen, a natural idea is dividing the private key into several parts. A general solution is to use (t, n) threshold secret sharing protocol [21, 26]. In (t, n) threshold secret sharing protocol, the secret is divided into n parts, and no participant less than $t-1$ can recover the complete

secret. Based on the above idea, many threshold based signature algorithms have been proposed [12, 22]. Even though threshold-based signature is a possible solution, there is still a potential risk here. Although the secret key is divided into n parts, the whole private key will still be re-established during the signature process. What's more, if the secret key is divided into too many parts, the signature generation time will be greatly increased.

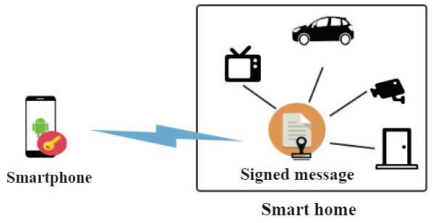


Figure 1: Traditional instructions transmission structure of smart home

To avoid these limitations, one common approach is dividing the private key into two parts, and storing them in different places (*i.e.* smartphone and smart device node). A valid signature generation negotiated by the smartphone and smart server. As shown Figure 2. It requires both parties to participate in the key generation phase and signature phase. Either party cannot recover the private key in the signature process.

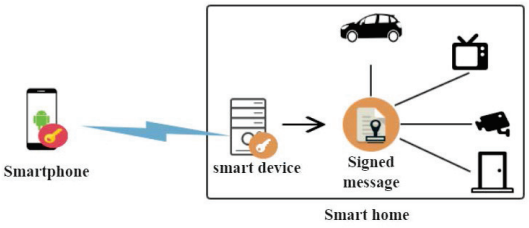


Figure 2: Two-part instructions transmission structure of smart home

To date, many two-party digital signatures have been proposed. A two-part DSA algorithm proposed by MacKenzie and Reiter [11] and the security of the algorithm was proved with random oracle model. Zhang [28] proposed a practical distributed two-party SM2 signature algorithm. SM2 algorithm is issued by the Chinese Government's State Cryptography Administration. A two-part ECDSA protocol proposed by Lindell [18], which is faster than previous protocols. But, many effective side-channel attacks for general NIST elliptic curve on smartphone has been published in [5,23]. He [14] and Zhang [27] proposed two-part identity-based signature protocol in 2018. However, identity based cryptography has a problem of key escrow.

For the sake of improving above deficiency, we present a two-party Edwards-curve digital signature algorithm. Edwards-curve digital signature algorithm developed by Daniel J. Bernstein *et al.*in 2012 [7]. In 2017, this digital signature formally defined in RFC 8032 [16] named EdDSA. EdDSA is more resilient to side-channel attacks, especially the cache-timing attack [3, 6]. The Edward curve [6] and parameter Curve25519 [8] has better efficiency and security than a general NIST elliptic curve [9, 25]. This algorithm is used widely in many softwares such as SSH, TLS, Tor, I2P, *etc.* and blockchains such as Monero, Naivecoin, Siacoin*etc.* [25].

1.1 Our Contribution

In this paper, a two-part siganture on edwards curve is proposed. A valid signature can be generated without reestablishing the whole private key. We implement our protocol on personal computers and mobile device (Android smartphone). Our scheme is more effective, practical and secure in mobile networks. We describe our contribution in detail as follows:

1) A two-part signature on edwards curve is proposed in this paper. According to our knowledge, this is the first two-part cryptography scheme designed for Edwards-curve digital signature algorithm without sacrificing security. This protocol generates a valid signature without reestablishing the whole private key. That is, either party cannot recover the private key in the signature process.

2) The security analysis of the protocol with standard assumptions is also presented. At the same time, in the signature generation phase, our protocol do not use random number generator, which can avoid the additional risks brought by random number generator.

3) This new protocol is implemented by us using Java(11.0.3) on PC and Android smartphone. The evaluation results demonstrate that our protocol has a faster speed in the secret key generation phase and smaller signature length compared with other protocols. Therefore, this scheme can be effectively deployed in practice to protect the private key security.

1.2 Organization of the Paper

This paper is organized as follows: Section 2, we review Edwards-curve digital signature scheme, paillier cryptosystem and zero-knowledge proof. Section 3, we present the protocol for two-part edwards-curve digital signature scheme. Section 4, we describe the security model of two-part Edwards-curve digital signature scheme. Section 5, we present the security analysis. Section 6, we implement our protocol using Java, and compared with other schemes. Section 7 we concludes.

2 Preliminaries

2.1 Edwards-Curve Digital Signature

In this section, we describe the Edwards-curve digital signature. This algorithm has four phases which include set

up, key generation, signature and verify.

2.1.1 EdDSA Setup

1) Construct a twisted Edwards curve over $\mathbb{F}_q$ as follows:

$$ax^2 + y^2 = 1 + dx^2y^2$$

$\mathbb{F}_q$ is a finite field where q is an odd prime. These two numbers a and b are described in [8]. If a point p satisfied this equation, then point $p\,(x_0, y_0)$ on the curve.

2) L is the number of points on the curve.

3) H is a hash function $H : \{0,1\}^* \to \{0,1\}^{512}$.

4) B is a generator of the elliptic curve group.

5) An integer b which is length of EdDSA public key.

6) Two functions $ENC(p)$ and $DEC(i)$. $ENC(p)$ can compress a point p to a integer. $DEC(i)$ can restore a compressed point from integer i.

2.2 EdDSA Key Generation

In this phase, the public key and private key is generated.

1) Choose an EdDSA private key which is a b-bit string k.

2) Calculate the hash $H(k) = (h_0, h_1, ..., h_{2b-1})$.

3) Define a scalar $s = 2^{b-2} + \sum_{3 \leqslant i \leqslant b-3} 2^i h_i$, compute $A = s \cdot B$.

4) Store the compressed value $ENC(A)$ as the public key.

5) The bits $h_b, ..., h_{2b-1}$ are used below during signing.

6) Output the key pair $(s, ENC(A))$.

2.2.1 EdDSA Signature

In this phase, signature (R, S) with message m is generated.

1) Define $r = H(h_b||...||h_{2b-1}||M)$.

2) Compute $R' = r \cdot B$.

3) Calculate $S = (r + H(ENC(R')||ENC(A)||H(M)) \cdot s) \bmod L$.

4) compute $R = ENC(R')$

5) Output EdDSA signature (R, S).

2.2.2 EdDA Verifiy

If following equation is satisfied, it means the signature is valid.

$$8S \cdot B = 8 \cdot R + 8Hash\,(R||ENC(A)||Hash(M)) \cdot A$$

or,

$$S \cdot B = R + Hash(R||ENC(A)||Hash(M)) \cdot A$$

2.3 Zero-Knowledge Proof

In this paper, we mainly use a technology ideal zero-knowledge functionality F_{zk}. The standard ideal zero knowledge functionality is defined by $((x, w), \lambda) \to (\lambda, (x, R(x, w)))$. λ is defined as empty string.

The zero knowledge functionality F_{zk}^R for relation R: Upon receiving $(prove, sid, x, w)$ from party $P_i (i \in 1, 2)$; if $(x, w) \notin R$ or sid has been previously used then ignore the message. Otherwise, send $(proof, sid, x)$to party P_{3-i} [18]. This zero-knowledge proof functionality can be implemented in the random oracle model [13].

The committed non-interactive ideal zero knowledge functionality F_{zk}^{R-com} for relation R. It works as follows: On receiving $(com\text{-}prove, sid, x, w)$ from a party $P_i (i \in 1, 2)$, ignore this message if $(x, w) \notin R$ or sid has been previously used. Store (sid, i, x) and send $(proof\text{-}receipt, sid)$ to P_{3-i}. Upon receiving $(decom\text{-}proof, sid)$ from a party $P_i (i \in 1, 2)$, if (sid, i, x) has been stored then send $(decom\text{-}proof, sid, x)$ to P_{3-i}.

In our two-part EdDSA protocol, we use the following ideal zero-knowledge functionalities: $F_{zk}^{R_p}$, $F_{zk}^{R_{DL}}$, $F_{zk}^{R_{PDL}}$, F_{zk}^{R-com}, which are defined in [18]. There are three relations here:

1) R_p is used to prove that a Paillier homomorphic encryption public key is successfully generated. $R_p = \{(N, (p_1, p_2))|N = p_1 \cdot p_2$ and p_1, p_2 are prime$\}$ It is a vaild Paillier public key, its proof in the [19].

2) Define the relation R_{DL} to prove knowledge of the discrete log of an Elliptic-curve point. $R_{DL} = \{(\mathbb{G}, G, q, P, w)|P = w \cdot G\}$ In our two-part EdDSA protocol, we use the Schnorr proof [20].

3) R_{PDL} is used to prove that a discrete log is encrypted by Paillier algorithm: $R_{PDL} = \{(c_{Paillier}, pk, Q_1, \mathbb{G}, q), (s_1, sk), s_1 = Dec_{sk}(c_{Paillier}), Q_1 = s_1 \cdot G$ and $s_1 \in \mathbb{Z}_q\}$. pk is a Paillier public key and sk is associated private key.

2.4 Paillier Encryption

In our two-part EdDSA protocol, we use Paillier encryption [19]. The Paillier encryption is defined as the following steps:

Key Generation.

1) Choose two random large prime numbers p and q with equal length.

2) Compute $n = pq, k = n + 1, \eta = \phi(n)$ and $\mu = \phi(n)^{-1} \bmod n$,where $\phi(n) = (p-1)(q-1)$.
3) The private key is $sk = (\eta, \mu)$ and the public key is $pk = (n, k)$.

Encryption.

1) Choose a message $m(0 \leqslant m \leqslant n)$ randomly.
2) Select a random v where $\mathbb{Z}^*_{n^2}$.
3) Calculate $c = k^m \cdot v^n \bmod n^2 = Enc_{pk}(m))$.

Decryption.
$m = L(c^\eta \bmod n^2) \cdot \mu \bmod n$, where $L(x) = \frac{x-1}{n}$.

Homomorphic Properties.
Define $c_1 = Enc_{pk}(m_1)$, $c_2 = Enc_{pk}(m_2)$, pk is public key, sk is private key.

1) $Enc_{pk}(m_1) \cdot Enc_{pk}(m_2) \bmod n^2 = Enc_{pk}(m_1 + m_2 \bmod n)$.
2) $Enc_{pk}(m_1)^{m_2} \bmod n^2 = Enc_{pk}(m_1 m_2 \bmod n)$.
3) $Enc_{pk}(m_1)^k \bmod n^2 = Enc_{pk}(km_1 \bmod n)$.

In this paper, we define homomorphic addition symbol is $\oplus$ and define homomorphic multiplication symbol is $\odot$. For example, $c_1 \oplus c_2 = Enc_{pk}(m_1 + m_2)$,$k \odot c_1 = Enc_{pk}(m_1)^k$.

3 Two-Part Edwards Curve Digital Signature

In this part, we show the two-part EdDSA signing protocoal. Two parties P_1, P_2 interact with each other to generate the public key and signature. The protocol has four phases which are setup, two-part key generation and two-part signing. The verify phase is same as EdDSA verify phase.

3.1 Setup

There are parameters:
$params = (a, d, q, L, B, H(x), b, ENC(p), DEC(i))$.
These parameters are the same with those in Section 2.1.

3.2 Two-Part Key Generation

In the two-part key generation phase, P_1 and P_2 interact with each other to generate the two-part EdDSA public key Q.

1) P_1 chooses a random string k_1 which is cryptographic security. Then, P_1 computes $str_1 = Hash(k_1)$. str_1 is the $P_1's$ private key generating elements. Then, P_1 computes itself private key $sl_1 = str_1[len]...str_1\left[\frac{len}{2}+1\right]$, which len is length of str_1. And then, P_1 gets itself signing parameters $perix1 = str_1[1]...str_1\left[\frac{len}{2}\right]$. P_1 computes $Q_1 = sl_1 \cdot B$. Then, P_1 sends $(prove\text{-}com, 1, Q_1, sl_1)$ to $F_{zk}^{R_{DL}-com}$. At last, P_1 generates a Paillier key-pair (pk, sk). P_1 sends $(prove, 1, N(p_1p_2))$ to $F_{zk}^{R_p}$, where $pk = N = p_1 \cdot p_2$.
2) When P_2 receives $(proof, 1, Q_1)$ from $F_{zk}^{R_{DL}}$ and $(proof, 1, N)$ from $F_{zk}^{R_p}$ successfully. P_2 chooses a random string k_2 which is cryptographic security. Then, P_2 computes $str_2 = Hash(k_2)$, str_2 is the private key generating elements of P_2. Then, P_2 computes itself private key $sl_2 = str_2[len]...str_2\left[\frac{len}{2}+1\right]$. The len is length of str_2. And then, P_2 gets its own signing parameters $perix2 = str_2[1]...str_2\left[\frac{len}{2}\right]$. P_2 computes $Q_2 = sl_2 \cdot B$. And then, P_2 sends $(prove, 2, Q_2, sl_2)$ to $F_{zk}^{R_{DL}}$. Finally, P_2 computes $Q = sl_2 \cdot Q_1$ and stores $(sl_2, Q, pk, perix2)$.
3) When P_1 receives $(proof, 2, Q_2)$ form $F_{zk}^{R_{DL}}$, P_1 computes $Q = sl_1 \cdot Q_2$, $Q' = ENC(Q)$. P_1 stores $(sl_1, Q, (sk, pk), perix1, Q')$, and P_1 sends $(decom, 1)$ to $F_{zk}^{R_{DL}}$. If not received , it abort.
4) Finally, if P_2 receives $(decom, 1, Q_1)$ from $F_{zk}^{R_{DL}}$, computes $Q = sl_2 \cdot Q_1$, $Q' = ENC(Q)$. P_2 stores $(sl_2, Q, pk, perix_2, Q')$. Otherwise, it abort. Obviously, $Q = sl_1 \cdot Q_2 = sl_2 \cdot Q_1 = (sl_1 \cdot sl_2)B$. Then P_1 and P_2 interaction processes are described in Figure 3.

3.3 Two-Part Signature Generation

In two-part signature generation phase, P_1 and P_2 interact with each other to generate the Two-part EdDSA Signature (R, S).

1) P_1 computes $r_1 = Hash(perix1||M) \bmod L$. Using Paillier algorithm encrypt r_1 and sl_1, such that $C_1 = Enc_{pk}(r_1)$, $C_2 = Enc_{pk}(sl_1)$. Then P_1 computes $Q_{r1} = r_1 \cdot B$. Finally, P_1 sends $(prove, 1, (c_1, c_2), (r_1, sk))$ and $(commit, 1, Q_{r1})$ to $F_{zk}^{R_{PDL}-com}$.
2) If P_2 receives $(proof, 1, (Q_r, C_1, C_2))$ and $(commit, 1)$ from $F_{zk}^{R_{PDL}-com}$,
then it executes the following steps ; otherwise, it aborts. P_2 computes $r_2 = Hash(perix2||M) \bmod L$. P_2 computes $Q_2' = ENC(Q_2)$; Then, P_2 computes $n_1 = Hash(perix2||Q_2') \bmod L$ and $n_2 = Hash(perix2||Q') \bmod L$. Then P_2 computes $e = Hash(m) \bmod L$ and $Q_{r2} = r_2 \cdot B$. P_2 sends $(prove, 2, R)$ to $F_{zk}^{R_{DL}}$.
3) When P_1 receives $(proof, 2, Q_2)$ from $F_{zk}^{R_{DL}}$, P_1 sends $(decom, 1)$ to $F_{zk}^{R_{DL}-com}$.
4) When P_2 receives Q_{r1}, P_2 computes part of signature $R = e \cdot r_2 \cdot Q_{r1}$ and $r = ENC(R)$. Next, P_2 computes $k = Hash(r||Q'||m) \bmod L$. According to the homomorphic properties of Paillier cryptosystem, P_2 can compute $s_1 = ((r_2 e) \bmod q + n_1 q) \odot C_1$,

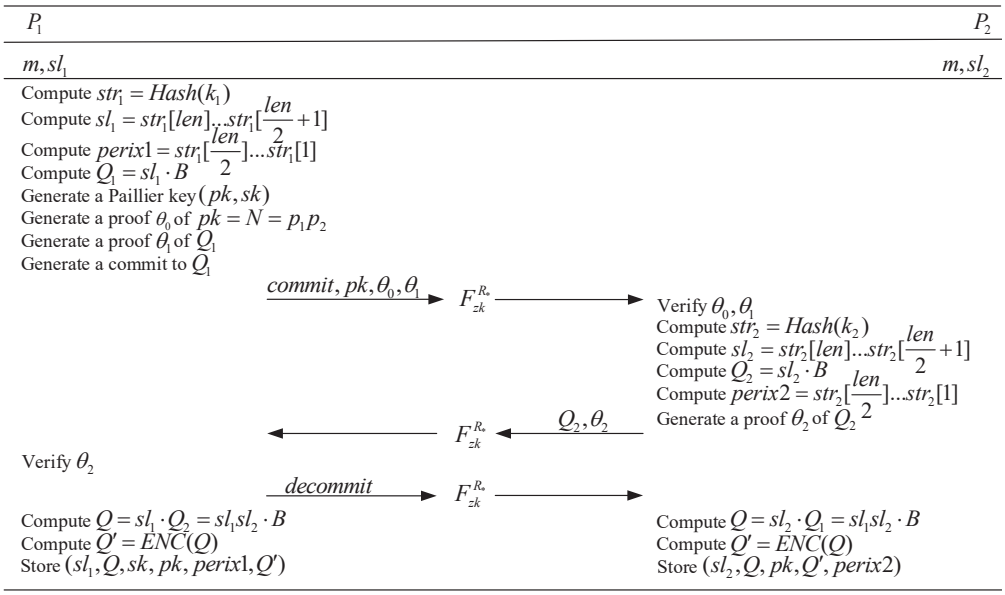


Figure 3: Two-part EdDSA key generation

$s_2 = ((ksl_2) \bmod q + n_2q) \odot C_2$. P_2 computes $s' = s_1 \oplus s_2$. Finally, P_2 sends s' to P_1.

5) If P_1 receives s' from $F_{zk}^{R_{DL}}$, then goes to the following steps; Otherwise, it abort. P_1 can compute the R, P_1 decrypts s' using its private such that $S = Dec_{sk}(s') \bmod L$. Finally, P_1 can verify and output the signature (R, S). Then P_1 and P_2 interaction processes are described in Figure 4.

3.4 Correctness

The correctness of the two-part EdDSA is proven as fellows:

$$\begin{aligned} S &= Dec(s') \bmod L = Dec(s_1 \oplus s_2) \bmod L \\ &= Dec_{sk}((((r_2e) \bmod L + n_1L) \odot c_1) \oplus \\ &\quad (((ksl_2) \bmod L + n_2L) \odot c_2)) \bmod L \\ &= ((r_2e) \cdot r_1) + ((ksl_2) \cdot sl_2) \bmod L \\ &= r_1r_2e + ksl_1sl_2 \bmod L \end{aligned}$$

4 Security Model

Definition 1. *Define a security of the digital signature scheme $\pi = (GenKey, Sign, Verify)$, we define an experiment $Sign\text{-}forge_{\mathcal{A},\pi}(1^h)$ consists of the following steps:*

1) $GenKey(1^h) \Rightarrow (pk, sk)$.

2) $\mathcal{A}^{Sign_{sk}(\cdot)}(1^h, pk) \Rightarrow (m^*, \sigma^*)$.

3) *$\mathcal{M}$ is the set of all message which can be queried. An adversary $\mathcal{A}$ can query the oracle to get message. Then, the experiment outputs 1 if $m^* \notin \mathcal{M}$ and $Verify(m^*, \sigma^*) = 1$.*

Definition 2. *If a signature scheme π is existentially unforgeable under chosen message attack for every probabilistic polynomial-time adversary $\mathcal{A}$, then there exist a negligible function η such that for every h.*

$$\Pr[Sign\text{-}forge_{\mathcal{A},\pi}(1^h) = 1] \leqslant \eta(h).$$

Then, we define an experiment $DistSign\text{-}forge^b_{\mathcal{A},\pi}(1^h)$. We define $\prod$ is our two-part signing protocol. In this experiment, an adversary $\mathcal{A}$ can control one of parties. $\prod_b(\cdot,\cdot)$ is an oracle which executes the instructions from party P_{3-b} honestly. The adversary $\mathcal{A}$ can interact with part P_{3-b} to generate signatures with any message. In this protocol, the key generation process is executed only once, and the signature generation process can be executed many times. The adversary $\mathcal{A}$ can query the oracle inputed two parameters , one is a session identifier (sid) and the other is input message or next incoming message. The oracle works as follows:

1) *Upon receiving a query $(0, 0)$ at the first time, P_{3-b} uses the oracle to initialize a machine M in the two-part key generation part of protocol $\prod$. If P_{3-b} sends the first message in this two-part key generation phase, then this message is the oracle's reply.*

2) *When oracle receives a query $(0, m)$, if the two-part key generation has not been completed, then oracle sends the message to the machine M as the next input message, and return M's reply. (If key generation phase has been finished, then oracle reply $\perp$ and exit.)*

3) *When oracle receives a query (sid, m), where $sid \neq 0$ and m is not the empty string. If two-part key generation phase is not completed, then the oracle returns $\perp$.*

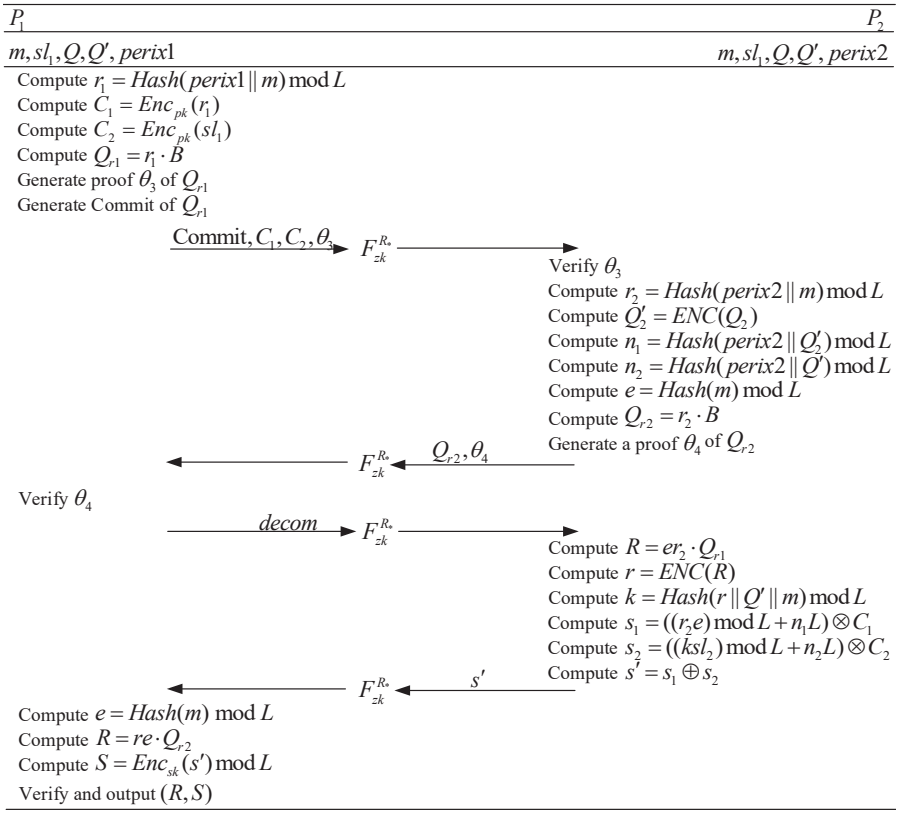


Figure 4: Two-part EdDSA siganture generation

4) *Upon receiving a query (sid, m), the two-part key generation phase has completed and this sid is the first time requested, that means this is a sign require. The oracle call a new machine M_{sid} running the instructions of part P_{3-b} in protocol $\prod$ with (sid, m). M_{sid} is initialized with the key and the parameters stored in M at the end of the two-part key generation phase. If party P_{3-b} sends the first message in the two-part signing phase, then the oracle replies with this message.*

5) *When oracle receives a query (sid, m), the two-part key generation has already finished and sid is not the first time queried, then the oracle sends m to M_{sid}. M_{sid} uses the message m as an input message. The oracle uses M_{sid}'s output as the next message. If M_{sid} completes, then M_{sid}'s output is returned.*

In the above process, an adversary can control part $P_b (b \in 1, 2)$ and can interact with $\prod_b$ in this experiment. If the adversary can forge a signature using a message which has not been queried to the oracle, then the adversary wins. We define the experiment $DistSign\text{-}forge^b_{\mathcal{A},\pi}(1^h)$ and the security definition of protocol $\prod$.

Definition 3. *We define an security experiment $DistSign\text{-}forge^b_{\mathcal{A},\pi}(1^h)$ as follows:*

1) $\mathcal{A}^{\prod_b(\cdot,\cdot)} \Rightarrow (m^*, \sigma^*)$.

2) *Let $\mathcal{M}$ be the set of all messages which adversary $\mathcal{A}$ could query. The adversary can query oracle with form (sid, m). If and only if $m^* \notin \mathcal{M}$ and $Verify_{pk}(m^*, \sigma^*) = 1$, then the experiment outputs 1, where the pk is two-part signing system verification key which output by P_{3-b} in two-part key generation. Verify is an algorithm of $\pi = \{GenKey, Sign, Verify\}$.*

Definition 4. *A protocol $\prod$ is a security two-part protocol for the two-part signature generation for scheme π, if for every probabilistic polynomial-time oracle machine adversary $\mathcal{A}$ and $b \in [1, 2]$, there exist a negligible function η such that for every h:*

$$\Pr[DistSign\text{-}forge^b_{\mathcal{A},\pi}(1^h) = 1] \leqslant \eta(h).$$

Definition 5. *We define a security function $Sign_{EdDSA}$, this function have two subroutines work in the security analysis of protocol $\prod$. These two subroutines as follows:*

1) $(x, Q) \leftarrow Expt\text{-}EdDSA_{keygen}(n)$.

2) $(R, S) \leftarrow Expt\text{-}EdDSA_{signing}(m)$.

Define $Expt\text{-}EdDSA_{keygen}(n)$ is EdDSA key generation function, which generate an EdDSA key pair (x, Q) by invoke the EdDSA key generation algorithm. We define n as a counter. Define $Expt\text{-}EdDSA_{signing}(m)$ is EdDSA singing function, which generate an EdDSA signature (S, R) by invoke the EdDSA signing algorithm. We define m is input message.

5 Security Analysis

In this section, we give the security analysis of our two-part protocol.

Theorem 1. *If the EdDSA signature is existentially-unforgeable under a chosen message attack and Paillier encryption scheme is indistinguishable under chosen-plaintext attack, then our two-part signature algorithm is secure.*

Proof. First of all, in our protocol we use zero-knowledge technology to proof data between P_1 and P_2.All parties can prove the authenticity of the data through zero-knowledge proof-of-knowledge. If adversary $\mathcal{A}$ can break these zero-knowledge model with probability ε, then it can break this two-part protocol with probability $\varepsilon \pm \eta(k)$ where $\eta(k)$ is a negligible function. □

In the process of proof, we assume adversary can corrupt P_1 or P_2 in the experiment. We prove respectively that P_1 corrupted and P_2 corrupted. In addition, we construct an adversary $\mathcal{A}_s$ who can forge a vaild EdDSA signature with probability μ_1 in Definition 1. The probability μ_1 is close to the probability that $\mathcal{A}$ forges a vaild signature in Definition 3 negligibly. Adversary $\mathcal{A}_s$ can invoke EdDSA key generation function and EdDSA signing function which describe in Definition 5 with input (sid, m).

Assume that Paillier has CPA security, then for any probabilistic polynomial time algorithm $\mathcal{A}$ and $b \in [1, 2]$ exists a probabilistic polynomial-time algorithm $\mathcal{A}_s$ and a negligible function μ for every h:

$$|\Pr[Sign\text{-}forge_{\mathcal{A},\pi}(1^h) = 1] - \Pr[DistSign\text{-}forge^b_{\mathcal{A},\prod}(1^h) = 1]| \leq \mu(h). \quad (1)$$

Where, π denotes EdDSA signature scheme and $\prod$ denotes the Two-part EdDSA signature scheme. If EdDSA is security, then according to Definition 2 there exists a negligible function μ' for every h, we can conclude $|\Pr[Sign\text{-}forge_{\mathcal{A},\pi}(1^h) = 1]| \leq \mu'(h)$. On the basis of Equation (1), we conclude that $|\Pr[DistSign\text{-}forge^b_{\mathcal{A},\prod}(1^h) = 1]| \leq \mu(h) + \mu'(h)$, obviously $\mu(h) + \mu'(h)$ is negligible, thus according to Definition 4 we conclude $\prod$ is security. Now, we just need to proof Equation (1) holds in the case of $b = 1$ and $b = 2$, respectively.

If $b = 1$, it means adversary $\mathcal{A}$ corrupts part P_1. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary in $DistSign\text{-}forge^b_{\mathcal{A},\prod}$. We construct a probabilistic polynomial-time adversary $\mathcal{A}_s$ in $Sign\text{-}forge^b_{\mathcal{A},\pi}$. $\mathcal{A}_s$ simulates the execution for $\mathcal{A}$ as follows:

1) First, adversary $\mathcal{A}_s$ can invoke $Expt\text{-}EdDSA_{keygen}(n)$ for Q. Q is the public key of EdDSA.

2) Second, $\mathcal{A}_s$ invokes $\mathcal{A}$ by inputing 1^h to simulate oracle $\prod$ for $\mathcal{A}$ in $DistSign\text{-}forge$ as follows:

a. At the beginning of the experiment, $\mathcal{A}$ always replies $\perp$ in the following two cases. The first case is that the key generation phase has not completed the second case is that the adversary $\mathcal{A}$ has not query $(0, 0)$ to $\prod$. After $\mathcal{A}$ queries $(0, 0)$ to $\prod$ that means this experiment is starting.

b. When $\mathcal{A}_s$ receives the first message $(0, m_1)$ from P_1. It is the first message in the two-part key generation phase. $\mathcal{A}_s$ computes the oracle's reply as follow steps:

   i. Adversary $\mathcal{A}_s$ can analyze message $(prove, 1, Q_1, sl_1)$ sent by P_1 to $F_{zk}^{R_{DL}-com}$.
   ii. Adversary $\mathcal{A}_s$ checks $Q_1 = sl_1 \cdot B$. If this equation holds, $\mathcal{A}_s$ calculates $Q_2 = (sl_1)^{-1} \cdot Q$. Otherwise, $\mathcal{A}_s$ chooses a random point as Q_2.
   iii. Adversary $\mathcal{A}_s$ sets the oracle's replay to be $(proof, 2, Q_2)$ and sends it to $\mathcal{A}$.
   iv. Adversary $\mathcal{A}_s$ can analyze message $(prove, 1, N, (p_1, p_2))$ sent by P_1 to $F_{zk}^{R_p}$.
   v. Adversary $\mathcal{A}_s$ checks the equation $N = p_1 \cdot p_2 = pk$, If this equation is not equal, abort.
   vi. Adversary $\mathcal{A}_s$ stores $perix2 = str_2[1] ... str_2\left[\frac{len}{2}\right]$.

c. Adversary receives the second message $(0, m_2)$ from P_1 and works as follows:

   i. $\mathcal{A}_s$ parses m_2 form of $(decom, 1)$ as $\mathcal{A}$ intends to send to $F_{zk}^{R_{DL}-com}$.
   ii. If $Q_1 \neq sl_1 \cdot B$, $\mathcal{A}_s$ generates the oracle response to P_2 and aborts.
   iii. $\mathcal{A}_s$ stores $(Q, perix2, sl_2, pk)$ and the two-part key generation phase is completed.

d. $\mathcal{A}_s$ receives a query (sid, m) where sid has not been queried yet. Adversary $\mathcal{A}_s$ could invoke function $Expt\text{-}EdDSA_{signing}(m)$ with input m. The function will return a vaild EdDSA signature (R, S). Then, $\mathcal{A}_s$ will reply to $\mathcal{A}$ as follows:

   i. When first message (sid, m) received is in the form of $(prove, commit, (C_1, C_2), (Q_1, sl_1), r_1)$. If $Q_{r1} = r_1 \cdot B$ and $C_1 = Enc_{pk}(r_1)$, $C_2 = Enc_{pk}(sl_1)$, then $\mathcal{A}_s$ sets $Q_{r2} = (r_1)^{-1} \cdot R$. Otherwise, $\mathcal{A}_s$ randomly selects a point as Q_{r2}. $\mathcal{A}_s$ sends the message $(proof, Q_{r2}, 2)$ to $\mathcal{A}$ that $\mathcal{A}$ expects to receive.
   ii. Upon receiving a message of the form $(decommit, 1)$ for the second time from $\mathcal{A}$. If $Q_1 \neq sl_1 \cdot B$, then exit. Otherwise, $\mathcal{A}$ chooses a random number $\rho \in \mathbb{Z}$, compute $S' = Enc_{pk}(S + \rho \cdot L)$, where S is received from function $Expt - EdDSA_{signing}(m)$ and then $\mathcal{A}_s$ sets the oracle reply S' to $\mathcal{A}$.

iii. Once $\mathcal{A}$ halts and outputs a pair (m^*, σ^*) , adversary $\mathcal{A}_s$ outputs (m^*, σ^*). At this point, we could prove that Equation (1) holds. Our ultimate purpose here is to prove that $\mathcal{A}$'s view in the simulation by $\mathcal{A}_s$ is identical to its view in a real execution.

In the two-part key generation phase, there is a difference between the simulation and the real execution. The difference is the generation of Q_{r2}. In the real execution, P_2 computes P_2's private key $sl_2 = str_2[len]...str_2\left[\frac{len}{2}+1\right]$. And then, P_2 computes $Q_2 = sl_2 \cdot B$. However, in the simulation, if $\mathcal{A}_s$ not exit, then $\mathcal{A}_s$ computes $Q_2 = (sl_1)^{-1} \cdot Q$, where Q is the public verification key received by $\mathcal{A}_s$ from key generation function $Expt\text{-}EdDSA_{keygen}(n)$. Thus, the distribution over $sl_2 \cdot B$ and $(sl_1) \cdot Q$ are identical. Obviously, the public key Q obtained by executing the simulation process is the same as that obtained by the function $Expt\text{-}EdDSA_{keygen}(n)$ and we can think of Q as random point.

Therefore, in this phase the $\mathcal{A}$'s view in simulation is identical to its view in a real execution.

In the two-part signing phase, there are two differences. The one difference is the generation of R. In the real execution, P_2 computes $r_2 = Hash(perix2||\ m) \bmod L$ and $Q_{r2} = r_2 \cdot B$. And then, P_1 computes $R = er_1 \cdot Q_{r2}$ or P_2 computes $R = er_2 \cdot Q_{r1}$. But in the simulation, P_2 computes $Q_{r2} = (r_1)^{-1} \cdot R$, where R is generated by signing function $Expt\text{-}EdDSA_{signing}(m)$. Obviously the distribution between $Q_{r2} = r_2 \cdot B$ and $Q_{r2} = (r_1)^{-1} \cdot R$ is identical. The other difference is the generation of S' which is the ciphertext of signature. In the simulation, S is an encryption of $S + \rho \cdot L$. However, in real execution, it is an encryption of $s = r_1 r_2 e + ksl_1 sl_2 + (n_1 r_1 + n_2 r_2)q$ where n_1,n_2 are hash value generated by P_2.

Observe that by the definition of EdDSA signature $s = r + kd = r_1 r_2 e + ksl_1 sl_2 \bmod L$, we can imply that $r_1 r_2 e + ksl_1 sl_2 = s \bmod L$, then $r_1 r_2 e + ksl_1 sl_2 = s + l \cdot L$. Therefore, the difference between the real execution and simulation with S is that:

1) Real execution: $s + l \cdot L + \rho \cdot L$.

2) Simulation: $s + \rho \cdot L$.

Since the distribution of S in the real execution and the simulation is statistically close. We prove that Equation (1) holds for $b = 1$.

If $b = 2$, we use a similar method described above, adversary $\mathcal{A}_s$ corrupts part P_2. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary in $DistSign\text{-}forge^{b}_{\mathcal{A},\prod}$ and $\mathcal{A}_s$ be a probabilistic polynomial-time adversary in $Sign\text{-}forge_{\mathcal{A},\pi}$.

In this phase simulation must be designed to work without knowing the paillier private key. That means $\mathcal{A}_s$ should not know the paillier private key. There is a potential problem here, P_2 may send the wrong value to P_1. In order to solve this problem, we assume that $\mathcal{A}_s$ will abort at some random point. $\mathcal{A}_s$ chooses $i \in [1, ..., p(h)+1]$ randomly. $p(h)$ is the upper bound of the number of query. If $\mathcal{A}_s$ chose correctly with probability $\frac{1}{p(h)+1}$ that means $\mathcal{A}_s$ could simulate $\mathcal{A}$'s view with probability $\frac{1}{p(h)+1}$. In this case, we consider S' is right. Thus, $\mathcal{A}$ can forge a signature in $Sign_{EdDSA}$ with probability at least $\frac{1}{p(h)+1}$. The probability of $\mathcal{A}_s$ can forge a signature in $\prod$ also at least $\frac{1}{p(h)+1}$.
The $\mathcal{A}_s$ work as follows:

1) Adversary $\mathcal{A}_s$ receives $(1^n, Q)$ from key generation function $Expt - EdDSA_{keygen}(n)$, where Q is a public key for EdDSA.

2) Let $p(h)$ be upper bound of the query times that $\mathcal{A}$ queries $\prod$, $\mathcal{A}_s$ chooses $i \in \{1, 2, ..., p(h)+1\}$ randomly.

3) $\mathcal{A}_s$ invokes $\mathcal{A}$ and simulates oracle $\prod$ in $DistSign\text{-}forge^{b=2}_{\mathcal{A},\prod}(1^h)$ as follows: In two-part key generation phase.

   a. Before the two-part key generation phase finished, $\mathcal{A}_s$ always replies $\perp$ to all queries. Whatever the content of the query is. Before $\mathcal{A}$ query $(0,0)$ to $\prod$, $\mathcal{A}_s$ always replies $\perp$ to all queries.

   b. After $\mathcal{A}$ queries $(0,0)$ to $\prod$, $\mathcal{A}_s$ generates a valid paillier encryption key-pair (sk, pk) and sets the oracle reply $(proof, 1, N)$ and $(commit, 1)$.

   c. Adversary $\mathcal{A}_s$ receives the first message $(0, m_1)$ from P_2. It is P_2's first message in the two-part key generation phase.

      i. Adversary $\mathcal{A}_s$ can analyze message $(prove, 2, Q_2, sl_2)$ sent by P_2 to $F_{zk}^{R_{DL}}$.

      ii. $\mathcal{A}_s$ checks $Q_2 = sl_2 \cdot B$. If this equation is not equal, then abort.

      iii. $\mathcal{A}_s$ sets the oracle reply to $(decom, 1, Q_1)$, where $Q_1 = (sl_1)^{-1} \cdot Q$.

      iv. Adversary $\mathcal{A}_s$ stores $perix1 = str_1[1]...str_1\left[\frac{len}{2}\right]$.

      v. $\mathcal{A}_s$ stores $(sl_1, pk, sk, perix1, Q)$ two-part key generation phase finished.

In the two-part signing phase:

4) Upon receiving a query of the form (sid, m) where sid has not been queried yet. $\mathcal{A}_s$ queries the Signing function $Expt - EdDSA_{sigiing}(m^*)$ with m and obtains a valid signature (R, S). Then, adversary $\mathcal{A}_s$ interacts with $\mathcal{A}$ in the following steps:

   a. $\mathcal{A}$ receives the first message $(proof, 2, Q_2)$ that $\mathcal{A}$ sends to $F_{zk}^{R_{DL}}$. $\mathcal{A}_s$ checks $Q_2 = r_2 c \cdot B$ and that Q_2 is a vaild point. $\mathcal{A}_s$ computes $Q_{r1} = (r_2)^{-1} \cdot B$. $\mathcal{A}_s$ sends $(decom, Q_1, 1)$ to P_2 as if coming from $F_{zk}^{R_{DL}}$.

b. When receiving second message (sid, m_2). The message m_2 is encrypted signature s'. If and only if this is the i-th call by $\mathcal{A}$ to the oracle $\prod$ it continues. Otherwise, it aborts.

5) Once $\mathcal{A}$ halts and outputs (m^*, σ), $\mathcal{A}_s$ outputs (m^*, σ) and halts.

When the (sid, m) is first query and it is the $j-th$ query. P_1 does not obtain a valid signature (R, S) with public verification key Q. We consider $i = j$, then the difference between the $\mathcal{A}$'s view in a real execution and the simulated execution by $\mathcal{A}_s$ is C_1 and C_2 . In the real execution, $C_1 = Enc_{pk}(sl_1)$, $C_2 = Enc_{pk}(r_1)$, $Q_1 = sl_1 \cdot B$ and $r_1 = Hash(perix1||M) \bmod L$, however in the simulation sl_1 and r_1 are random value, it is identical. Since $\mathcal{A}_s$ has not hold the pailier private key in the simulation. The indistinguishability of the simulation follows from a reduction of indistinguishability of the encryption scheme under CPA. We can conclude that:

$$|\Pr[Sign\text{-}forge_{\mathrm{A},\pi}(1^h) = 1 | i = j] - \Pr[DistSign\text{-}forge^2_{\mathrm{A},\prod}(1^h) = 1]| \leqslant \eta(h).$$

$$\Pr[DistSign\text{-}forge^2_{\mathrm{A},\prod}(1^h) = 1] \leqslant \frac{\Pr[Sign\text{-}forge_{\mathcal{A}_s,\pi}(1^h) = 1]}{p(h)+1} + \eta(h).$$

$$\Pr[Sign\text{-}forge_{\mathrm{A},\pi}(1^h) = 1] \geq \frac{\Pr[DistSign\text{-}forge^2_{\mathrm{A},\prod}(1^h) = 1]}{p(h)+1} - \eta(h).$$

We can conclude that if adversary $\mathcal{A}$ forges a valid signature in $Sign\text{-}forge_{\mathrm{A},\pi}$ with a non-negligible probability, then $\mathcal{A}_s$ can forge a valid signature in $DistSign\text{-}forge^{b=2}_{\mathrm{A},\prod}$ with a non-negligible probability.

6 Efficiency and Experimental Results

In this section, we will give a comparison of our work with other protocols and show the experiment results.

We implemented our protocol, Zhang's [28] protocol, Lindell's [18] protocol and Zhang's [27] protocol, respectively, using Java(11.0.3) and Jpbc-lib on personal computer with Intel i5-4200H @2.80Ghz processor, 8G bytes memory and Microsoft Windows 10 x64 professional operating system. In [28], [18] and our protocol we use curve25519. We also implemented these four protocols on Android devices(HUAWEI nova3e with Kirin 659 2.36 Ghz processor, 4G bytes memory and Android 9 operating system).

First of all, we compare the running time of the four protocols in different phases on PC and smartphone. The results are illustrated in Figure 5 and Figure 6, respectively, which were obtained in 30 executions. The experimental results show that our protocol runs faster than the other two algorithms [18, 28] in the key generation phase under the same elliptic curve conditions. The two-part identity-based signature scheme [27] runs faster in key generation phase and sign phase. However, it takes more time in the setup phase and have a problem of key escrow.

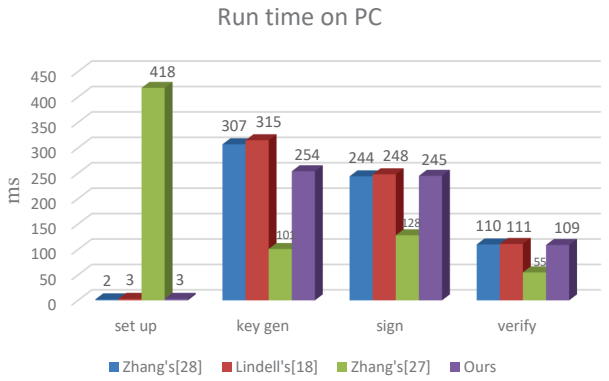


Figure 5: Run time on PC

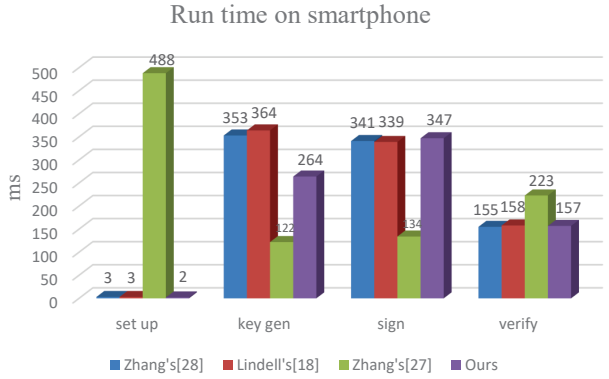


Figure 6: Run time on Smartphone

Secondly, we analyzed the efficiency of each stage of our scheme on PC. As shown in Figure 7. In two-part key generation phase, Keygen_phase1 is defined as the execution process of P_1 before the first message, Keygen_phase2 as the execution process of P_2 between the first message and the second message, and Keygen_phase3 as the execution process after the second message. In two-part signing phase, Sign_phase1 is defined as the execution process of P_1 before the first message, Sign_phase2 as the execution process of P_2 between the first message and the second message, Sign_phase3 as the execution process of P_2 between the second message and the fourth message, and Sign_phase4 as the execution process of P_1 after the fourth message.

Then, we compare the storage space of these three signature protocols in Table 1. Our public key length is one byte shorter than those of other two schemes. Our signature length is 8 bytes shorter than that in Lindell's protocol [18].

Finally, we analyze the size of message as follows. In the two-part key generation phase, the length of all messages of P_1 is $256bits + 1024bits + 512bits = 1792bits$.

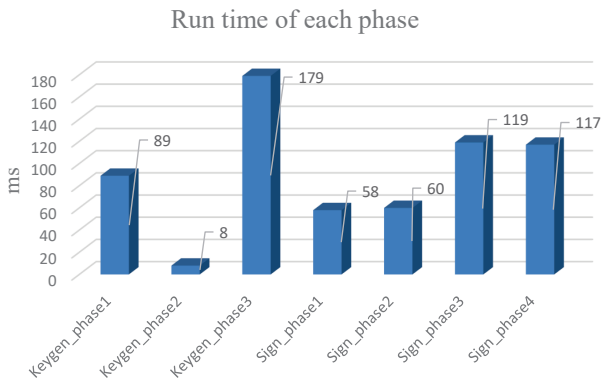


Figure 7: Run time of each phase

Table 1: Length of public key and signature

Protocol	*Public key*	*Signature*
Zhang's [28]	33 bytes	64 bytes
Lindell's [18]	33 bytes	72 bytes
Ours	32 bytes	64 bytes

The length of all messages of P_2 is $256bits$. In the two-part signing phase, the length of all message of P_1 is $256bits + 1024bits + 1024bits = 2304bits$, the length of all messages of P_2 is $2048bits + 256bits = 2304bits$. Therefore, it is efficient for practical application even in restricted environment.

7 Conclusions

- In this paper, we have proposed a two-part signature on edwards curve. This protocol could be effectively applied in blockchain to reduce the risk of key loss. We describe the steps of this protocol and give an analysis of security. Further, we implemented our algorithm and compare with other schemes.
- This protocol can be applied to many real-world scenarios. For example, the signature of blockchain transactions. The security authentication of users in the Internet of Things and Internet of Vehicle*etc.*
- In the future work, we need implement algorithm in real-world applications to further obtain operational data and improve the real execution efficiency of the algorithm.

Acknowledgments

This research was supported by National Key R and D Program of China (No.2017YFB0802000), the National Natural Science Foundation of China (61802241, 61572303, 61772326, 61802242), the National Cryptography Development Fund during the 13th Five-year Plan Period (MMJJ20180217), the Foundation of State Key Laboratory of Information Security (2017-MS-03), the Fundamental Research Funds for the Central Universities (GK201702004).

References

[1] A. A. Al-khatib and W. A. Hammood, "Mobile malware and defending systems: Comparison study," *International Journal of Electronics and Information Engineering*, vol. 6, no. 2, pp. 116–123, 2017.

[2] W. Ali, G. Dustgeer, M. Awais, and M. A. Shah, "Iot based smart home: Security challenges, security requirements and solutions," in *The 23rd International Conference on Automation and Computing (ICAC'17)*, pp. 1–6, 2017.

[3] K. M. Alonso, *Monero-Privacy in the Blockchain*, 2018. (`https://eprint.iacr.org/2018/535.pdf`)

[4] N. Balta-Ozkan, B. Boteler, and O. Amerighi, "European smart home market development: Public views on technical and economic aspects across the united kingdom, germany and italy," *Energy Research & Social Science*, vol. 3, pp. 65–77, 2014.

[5] P. Belgarric, P. A. Fouque, G. Macario-Rat, and M. Tibouchi, "Side-channel analysis of weierstrass and koblitz curve ECDSA on android smartphones," in *Cryptographers' Track at the RSA Conference*, pp. 236–252, 2016.

[6] D. J. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters, "Twisted edwards curves," in *International Conference on Cryptology in Africa*, pp. 389–405, 2008.

[7] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B. Y. Yang, "High-speed high-security signatures," *Journal of Cryptographic Engineering*, vol. 2, no. 2, pp. 77–89, 2012.

[8] D. J. Bernstein, S. Josefsson, T. Lange, P. Schwabe, and B. Y. Yang, "Eddsa for more curves," *Cryptology ePrint Archive*, pp. 5, vol. 2015, 2015. (`https://ed25519.cr.yp.to/eddsa-20150704.pdf`)

[9] D. J. Bernstein and T. Lange, "Failures in Nist's ECC standards," 2016. (`https://cr.yp.to/newelliptic/nistecc-20160106.pdf`)

[10] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *IEEE Symposium on Security and Privacy (SP'16)*, pp. 636–654, 2016.

[11] R. Gennaro, S. Goldfeder, and A. Narayanan, "Two-party generation of DSA signatures," in *Annual International Cryptology Conference*, pp. 137–154, 2001.

[12] R. Gennaro, S. Goldfeder, and A. Narayanan, "Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security," in *International Conference on Applied Cryptography and Network Security*, pp. 156–174, 2016.

[13] C. Hazay and Y. Lindell, "Efficient secure two-party protocols: Techniques and constructions," *Information Security and Cryptography*, 2010. ISBN: 978-3-642-14303-8.

[14] D. He, Y. Zhang, D. Wang, and K. K. R. Choo, "Secure and efficient two-party signing protocol for the identity-based signature scheme in the IEEE P1363 standard for public key cryptography," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2018. DOI: 10.1109/TDSC.2018.2857775.

[15] M. S. Hwang, E. F. Cahyadi, C. Y. Yang, and S. F. Chiou, "An improvement of the remote authentication scheme for anonymous users using an elliptic curve cryptosystem," in *IEEE 4th International Conference on Computer and Communications (ICCC'18)*, pp. 1872–1877, 2018.

[16] S. Josefsson and I. Liusvaara, *Edwards-Curve Digital Signature Algorithm (EdDSA)*, RFC 8032, 2017.

[17] K. Karimi and S. Krit, "Smart home-smartphone systems: Threats, security requirements and open research challenges," in *International Conference of Computer Science and Renewable Energies (ICCSRE'19)*, pp. 1–5, 2019.

[18] Y. Lindell, "Fast secure two-party ECDSA signing," in *Annual International Cryptology Conference*, pp. 613–644, 2017.

[19] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, 1999.

[20] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

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

[22] H. M. Sun, N. Y. Lee, and T. Hwang, "Threshold proxy signatures," *IEE Proceedings-Computers and Digital Techniques*, vol. 146, no. 5, pp. 259–263, 1999.

[23] S. Takarabt, A. Schaub, A. Facon, S. Guilley, L. Sauvage, Y. Souissi, and Y. Mathieu, "Cache-timing attacks still threaten IoT devices," in *International Conference on Codes, Cryptology, and Information Security*, pp. 13–30, 2019.

[24] C. Y. Tsai, C. S. Pan, and M. S. Hwang, "An improved password authentication scheme for smart card," in *International Conference on Intelligent and Interactive Systems and Applications*, pp. 194–199, 2016.

[25] L. Wang, X. Shen, J. Li, J. Shao, and Y. Yang, "Cryptographic primitives in blockchains," *Journal of Network and Computer Applications*, vol. 127, pp. 43–58, 2019.

[26] C. C. Yang, T. Y. Chang, and M. S. Hwang, "A (t, n) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, 2004.

[27] Y. Zhang, D. He, S. Zeadally, D. Wang, and K. K. R. Choo, "Efficient and provably secure distributed signing protocol for mobile devices in wireless networks," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5271–5280, 2018.

[28] M. Zhang, Y. Zhang, D. He, "A provable-secure and practical two-party distributed signing protocol for SM2 signature algorithm," *Frontiers of Computer Science*, vol. 14, no. 3, pp. 1–14, 2020.

Biography

Mingrui Zhang received the B.S degree from Xi'an University of Posts & Telecommunications in 2018, now he is a M.S. degree candidate in School of Computer Science, Shaanxi Normal University, Xi'an, China. His research interests include secure network protocols and its analysis.

Bo Yang received the B.S. degree from Peking University in 1986, and the M.S. and Ph.D. degrees from Xidian University in 1993 and 1999, respectively. From July 1986 to July 2005, he had been at Xidian University. From 2002, he had been a professor of National Key Lab. of ISN in Xidian University. He has served as a program chair for the fourth China Conference on Information and Communications Security in 2005, the vice-chair for ChinaCrypt 2009, and the general co-chair for the Joint Workshop on Information Security since 2010. He is currently a professor and supervisor of Ph.D. candidates at the School of Computer Science, Shaanxi Normal University, a Bai-Ren project special-term professor of Shaanxi Province, and a member of the Council of Chinese Association for Cryptologic Research. His research interests include information theory and cryptography.

Hongxia Hou Ph.D. candidate, now she is an associate professor in Xi'an University of Posts & Telecommunications. Her main research interests include cryptography and information security.

Meijuan Huang Ph.D. candidate, now she is an associate professor in Baoji University of Arts and Sciences. Her main research interests include cryptography.

Yanwei Zhou is an associate professor in Shaanxi Normal University. His main research interests include cryptography and information security.

Elliptic Curve Scalar Multiplication Algorithm Based on Side Channel Atomic Block over GF(2^m)

Shuang-Gen Liu, Yan-Yan Hu, and Lan Wei
(Corresponding author: Shuang-Gen Liu)

School of Cyberspace Security, Xi'an University of Posts and Telecommunications
Xi'an 710121, China
Email: liusgxupt@163.com

(Received Feb. 28, 2020; Revised and Accepted Dec. 10, 2020; First Online May 30, 2021)

Abstract

Elliptic Curve Cryptography (ECC) has become one of the research hotspots in cryptography in recent years. Scalar multiplication is the most crucial operation in ECC, and it largely determines the efficiency of ECC. To improve ECC's speed, we propose a new secure and efficient scalar multiplication algorithm on elliptic curves over GF(2^m). In addition, we present the new composite operation formulas $3P_1$ and $2P_1 + P_2$ using only x-coordinate, where P_1 and P_2 are points on an elliptic curve. To ensure the safety and efficiency of the proposed algorithm, we constitute an atomic block by adding dummy operations and using the Montgomery trick.

Keywords: Elliptic Curve Cryptography; Scalar Multiplication; Side Channel Atomic Block; Simple Power Analysis

1 Introduction

As the hacking techniques become more and more powerful, safe and efficient encryption technology is needed. Since Miller [17] and Koblitz [9] independently proposed Elliptic Curve Cryptography (ECC) in 1985, it has become one of the research hotspots in the field of cryptography due to its short key and high security. ECC can provide the same functions as the RSA cryptosystem and it requires a shorter key length than RSA under the same security. It is generally used for digital signature, authentication, encryption, decryption [8, 19, 25]. Because of its advantages in security, encryption and decryption performance, and space consumption, ECC has a wide range of applications, such as transport layer security (TLS), cryptocurrency, SM2 public key cryptography and government agencies, etc. Besides, it is especially suitable for environments with limited storage resources, such as smart cards and secure storage chips.

In ECC, it is easy to obtain the point Q when $Q = kP$, and the number k and point P are given. But it is difficult to find k when point P and point Q are given. This is the classical discrete logarithm problem (DLP). ECC uses this feature to encrypt where point Q is the public key, big number k is the private key and point P is the base point on an elliptic curve. The most crucial operation in ECC is scalar multiplication kP that largely determines the speed of ECC. There are two main methods to improve the efficiency of scalar multiplication. The first method is to reduce computation by optimizing the bottom arithmetic formulas, such as reducing the number of field inversion operations by transforming coordinates. The second method is to decrease the number of point addition and doubling in the scalar multiplication algorithm by studying the expanded form of k, such as double-base chain [27] and symmetric ternary form (STF) [13].

Side channel analysis (SCA) is a method to attack the cryptographic devices by analyzing the leaked side channel information such as time consumption, power consumption or electromagnetic radiation during the operation of cryptographic devices [24]. Power analysis is a form of SCA. It is an attack by collecting power consumption information generated by cryptographic devices or cryptographic chips during encryption, decryption or signature operations, and analyzing the key by using statistics, cryptography and other relevant knowledge. Power analysis can be divided into simple power analysis (SPA) and differential power analysis (DPA). SPA has a direct threat to cryptographic devices. It can directly analyze the power information collected during the execution of cryptographic algorithm. When the device performs encryption or decryption, the key can be derived from the difference in power consumption trajectories. The key in this paper refers to the private key k.

In 1987, the Montgomery algorithm was proposed by Montgomery [18]. The basic idea is that each loop has a point addition and doubling so that the energy consumed

by each loop is basically the same. In 1999, Lopez and Dahab [16] optimized the Montgomery ladder algorithm on elliptic curves over GF(2^m). The new point addition and doubling formulas eliminated the calculation of y-coordinate, which improved the calculation speed of the algorithm. In 2008, the new point addition and doubling formulas proposed by Yu *et al.* [**?**] not only omitted the y-coordinate but also dislodged the field inversion operation. In 2013, Sung *et al.* [5] posed the new composite formulas $4P_1$, $3P_1 + P_2$ and $2P_1 + 2P_2$ with only x-coordinate, and presented the extended quaternary Montgomery ladder algorithm over GF(2^m). In 2016, Lai and Zhang [10] proposed Co_Z point addition algorithm, conjugate point addition algorithm and point doubling-point addition algorithm with omitting Z-coordinate on Hessian elliptic curves and applied them to the traditional Montgomery ladder algorithm. In 2017, Yu *et al.* [26] optimized the Montgomery algorithm using the Co_Z technique in projective coordinates over GF(3^m). In 2019, Liu *et al.* [14] proposed the ternary Montgomery ladder algorithm, which combines the original Montgomery ladder algorithm with the ternary representation of the scalar k.

To obtain a safe and efficient scalar multiplication algorithm, we first propose the new composite operation formulas $3P_1$ and $2P_1 + P_2$ using only x-coordinate in affine coordinate system to reduce the bottom field operations and we apply them to the ternary Montgomery ladder algorithm. Then we constitute an atomic block by adding dummy operations to the proposed composite operation formulas to prevent SPA. Last, we use Montgomery trick in the atomic block to optimize the computational cost, which can decrease the number of field inversion operations.

The rest of this paper is presented as follows. In section II, we briefly introduce Elliptic Curve Cryptography and the Montgomery ladder algorithm. In section III, we give a detailed presentation on new composite operation formulas and the anti-SPA scalar multiplication algorithm based on side channel atomic block. In section IV, we compare the performance of the proposed algorithm with existing algorithms.

2 Elliptic Curve Cryptography and Montgomery Ladder Algorithm

2.1 Elliptic Curve Cryptography

Definition 1. *The equation of a non-super singular elliptic curve E over GF(2^m) is given as follows:*

$$E/GF(2^m) : y^2 + xy = x^3 + ax^2 + b. \tag{1}$$

with $a, b \in GF(2^m), b \neq 0$. All points on E and the infinity point $\mathcal{O}$ form an abelian group. Assume $P_1 = (x_1, y_1) \in E(GF(2^m))$, $P_2 = (x_2, y_2) \in E(GF(2^m))$, $-P_1 = (x_1, x_1 + y_1)$ and $P_2 \neq -P_1$.

If $P_1 \neq P_2$, $P_3 = P_1 + P_2 = (x_3, y_3)$, then point addition operation:

$$\begin{cases} x_3 = (\dfrac{y_1 + y_2}{x_1 + x_2})^2 + \dfrac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a \\ y_3 = \dfrac{y_1 + y_2}{x_1 + x_2}(x_1 + x_3) + x_3 + y_1 \end{cases} \tag{2}$$

If $P_1 = P_2$, $P_3 = 2P_1 = (x_3, y_3)$, then point doubling operation:

$$\begin{cases} x_3 = (x_1 + \dfrac{y_1}{x_1})^2 + \dfrac{y_1}{x_1} + x_2 + a \\ y_3 = (x_1 + \dfrac{y_1}{x_1})(x_1 + x_3) + x_3 + y_1 \end{cases} \tag{3}$$

It can be seen that the computational costs of point addition and doubling are both $1I + 2M + 1S$, where I, M, S are the representations of field inversion, field multiplication and field squaring, respectively.

2.2 Montgomery Ladder Algorithm

The Montgomery algorithm was initially proposed to improve the speed of scalar multiplication. The left-to-right Montgomery ladder algorithm [20] is described by Algorithm 1, which is a classical way to compute the scalar multiplication.

Algorithm 1 Left-To-Right Montgomery Ladder Algorithm

1: **Input:** $P = (x, y) \in E(GF(2^m))$, and $k = (k_{n-1}k_{n-2}\cdots k_1k_0)_2$
2: **Output:** $Q = kP \in E(GF(2^m))$
3: $R_0 = P, R_1 = 2P$
4: **for** $i \leq n-2, \cdots, 0$ **do**
5: **if** $k_i = 1$ **then**
6: $R_0 = R_0 + R_1, R_1 = 2R_1$
7: **else if** $k_i = 0$ **then**
8: $R_1 = R_0 + R_1, R_0 = 2R_0$
9: **end if**
10: **end for**
11: **Return** $Q = R_0$
12: End

Based on the original Montgomery ladder algorithm, Liu *et al.* [14] proposed the ternary Montgomery ladder algorithm, which is described by Algorithm 2.

3 New Algorithm Based on the Ternary Montgomery Ladder Algorithm

3.1 Composite Operation Formulas

Improving the performance of the Montgomery ladder algorithm by using only x-coordinate method was first

Algorithm 2 The Ternary Montgomery Ladder Algorithm

Input: $P = (x, y) \in E(GF(2^m))$, and $k = (k_{n-1}k_{n-2}\cdots k_1k_0)_3$, where $k_{n-1} = 1$ or 2
Output: $Q = kP \in E(GF(2^m))$
$R_0 = k_{n-1}P, R_1 = (k_{n-1}+1)P$
for $i \leq n-2, \cdots, 0$ **do**
  if $k_i = 0$ **then**
    $R_2 = 3R_0, R_1 = 2R_0 + R_1$
  else if $k_i = 1$ **then**
    $R_2 = 2R_0 + R_1, R_1 = 2R_1 + R_0$
  else if $k_i = 2$ **then**
    $R_2 = 2R_1 + R_0, R_1 = 3R_1$
  end if
  $R_0 = R_2$
end for
Return $Q = R_0$
End

introduced by Lopez & Dahab [16]. Then several x-coordinate-only methods were presented [5, 22, 28]. Assume P_i is a point on an elliptic curve E. Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, $-P_1 = (x_1, x_1 + y_1)$, $P_2 - P_1 = P = (x, y)$, and $P_2 \neq -P_1$, then we can obtain

$$x_3 = \begin{cases} x + \frac{x_1}{x_1+x_2} + \left(\frac{x_1}{x_1+x_2}\right)^2 & P_1 \neq P_2 \\ x_1^2 + \frac{b}{x_1^2} & P_1 = P_2 \end{cases} \quad (4)$$

The formula for restoring the y coordinate at the last step is

$$y_1 = (x_1 + x)\{(x_1 + x)(x_2 + x) + x^2 + y\}/(x + y) \quad (5)$$

It can be seen from Equation (4) that the costs of both two operations are $1I + 1M + 1S$. Based on the idea of Lopez & Dahab, this paper proposes two composite operation formulas $3P_1$ and $2P_1 + P_2$.

Theorem 1. *Let $P_1 = (x_1, y_1)$ be a point on an elliptic curve E over $GF(2^m)$. Then, x_{3P_1} can be gained:*

$$x_{3P_1} = x_1 + \frac{{x_1}^3}{{x_1}^4 + {x_1}^3 + b} + \left(\frac{{x_1}^3}{{x_1}^4 + {x_1}^3 + b}\right)^2 \quad (6)$$

with cost $1I+1M+3S+1C$, where C is the representation of field cubing.

Proof. Let $3P_1$ be computed as $2P_1 + P_1$. Equation (4) gives

$$x_{3P_1} = x_1 + \frac{x_{P_1}}{x_{P_1} + x_{2P_1}} + \left(\frac{x_{P_1}}{x_{P_1} + x_{2P_1}}\right)^2 \quad (7)$$

Then, we obtain Equation (6). □

Theorem 2. *Let $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ be points on an elliptic curve E over $GF(2^m)$. Then, $x_{2P_1+P_2}$ can be gained:*

$$x_{2P_1+P_2} = x_2 + \frac{x_1(x_1+x_2)^2}{(x+x_1)(x_1+x_2)^2 + x_1x_2} + \left(\frac{x_1(x_1+x_2)^2}{(x+x_1)(x_1+x_2)^2 + x_1x_2}\right)^2 \quad (8)$$

with cost $1I + 2S + 4M$.

Proof. Let $2P_1 + P_2$ be computed as $(P_1 + P_2) + P_1$ and $P_2 - P_1 = P(x, y)$ which is an input. Equation (4) gives

$$x_{2P_1+P_2} = x_2 + \frac{x_{P_1}}{x_{P_1} + x_{P_1+P_2}} + \left(\frac{x_{P_1}}{x_{P_1} + x_{P_1+P_2}}\right)^2 \quad (9)$$

Then, we obtain Equation (8). □

The probability that k_i is equal to 0, 1, and 2 is 1/3 [11]. When Algorithm 2 is computed by Equation (6) and Equation (8), the average calculation costs are $2I + 6M + 14/3S + 2/3C$ per loop.

3.2 New Algorithm Based on Side Channel Atomic Block

In view of SCA attack, in 2004, Mames, ciet and joye proposed a method that almost does not increase the amount of computation: Side channel atomic block method [4]. Its main idea is to decompose the operations on elliptic curves into a series of indistinguishable atomic blocks with multiple side channels. The general method is to add dummy operations so that there is no difference in the side channel analysis of different execution processes.

In this paper, as can be seen from Algorithm 2 that the discrimination of each loop is $3P_1$ and $2P_1 + P_2$. To make it anti-SPA, we can add some dummy operations to $3P_1$ and $2P_1 + P_2$ to make the costs of $3P_1$ and $2P_1 + P_2$ indistinguishable so that the amount of calculations in each loop is exactly the same. In this way, the scalar multiplication can be represented as a series of indistinguishable atomic blocks of code, so the attacker cannot obtain the side channel information by SPA attack.

The Montgomery trick is an efficient way to improve performance by computing field inversions simultaneously. For instance, a^{-1} and b^{-1} can be computed as $a^{-1} = (ab)^{-1} \cdot b$, $b^{-1} = (ab)^{-1} \cdot a$. It converts two field inversion operations into one field inversion operation and three field multiplication operations, which can save $1I - 3M$ calculation costs per loop. Therefore, we apply the Montgomery trick to Equation (6) and Equation (8) in the atomic block to reduce the amount of field inversion operations to optimize the algorithm.

As stated above, we constitute the atomic block by adding dummy operations in each loop to make it anti-SPA and using Montgomery trick to reduce the amount of field inversion operations. Table 1, called the atomic block elliptic curve triple and double-and-add, *i.e.* AETD, describes the atomic block in detail.

In the upper section, Algorithm 2 can be computed efficiently by using the proposed composite operation formulas Equation (6) and Equation (8), with cost $2I+14/3S+6M+2/3C$ per loop. However, the computation costs of AETD just require $1I+4S+10M$. Applying AETD to Algorithm 2, Algorithm 3 is obtained. Algorithm 3 saves $I+2/3S-4M+2/3C$ compared with Algorithm 2 computed by Equation (6) and Equation (8), and saves $3I-2M$ compared with Algorithm 2 computed by Equation (2) and Equation (3) in each loop. From Algorithm 3, we can conclude that only one atomic block is used in each loop, so each loop requires the same amount of calculations regardless of the value of k_i.

Algorithm 3 Anti-SPA Scalar Multiplication Algorithm Based On Side Channel Atomic Block

1: **Input:** $P=(x,y)\in E(GF(2^m))$, and $k=(k_{n-1}k_{n-2}\cdots k_1k_0)_3$, where $k_{n-1}=1$ or 2
2: **Output:** $Q=kP\in E(GF(2^m))$
3: $R_0=k_{n-1}P, R_1=(k_{n-1}+1)P$
4: **for** $i\le n-2,\cdots,0$ **do**
5: $(R_0,R_1)=AETD[k_i](R_0,R_1)$
6: **end for**
7: **Return** $Q=R_0$
8: End

4 Performance Analysis

4.1 Security Analysis

In ECC, if a scalar multiplication algorithm has different power consumption according to k_i, it is vulnerable to SPA. In other words, if the algorithm has the same power consumption regardless of the value of k_i, it is resistant to SPA. Therefore, all countermeasures against SPA have to modify the algorithm to obtain a uniform power consumption trace. In general, there are three main ways to anti-SPA. The first way is uniform algorithm behavior, such as Montgomery ladder algorithm. The second way is uniform point addition and doubling formulas, such as Edwards curve [2]. The third way is to add dummy field operations [6].

To improve the efficiency of the ternary Montgomery ladder algorithm, the composite operation formulas $3P_1$ and $2P_1+P_2$ are proposed. However, the power consumption of $3P_1$ and $2P_1+P_2$ is different. Algorithm 2 computes $3P_1$ and $2P_1+P_2$ when k_i is equal to 0 or 2, while it computes $2P_1+P_2$ twice when k_i is equal to 1. SPA gains the key according to the peak shape of the energy graph [12], so it is easy to obtain the value of k_i by observing the power consumption curve leaked during execution of the algorithm. Therefore, we adopt the third way to add dummy field operations to constitute an atomic block. It can be seen from Table 1 and Algorithm 3 that the field operation of each step of every atomic block is the same and only one atomic block is used in each loop, so the power consumed by each loop is the same whatever $k_i=0$, 1, or 2, which is secure to resist SPA. In addition, Algorithm 3 can also resist DPA so long as randomize the scalar k.

4.2 Efficiency Analysis

Because the extra calculations of algorithms are negligible, in this paper, we mainly compare the calculations of main iteration of algorithms. In this section, the efficiency of the proposed composite operation formulas and Algorithm 3 is analyzed.

Table 2 shows the computation costs of Algorithm 2 under different calculation formulas. From it, we can draw the conclusion that Algorithm 2 can be computed efficiently by using Equation (6) and Equation (8) proposed in this paper. It requires $2I+14/3S+6M+2/3C$ on average, with saving $2I-2M-2/3S-2/3C$ than Equation (4) and saving $2I+2M-2/3S-2/3C$ than Equation (2) and Equation (3) in each loop.

Given an integer k, assume that $m=\lceil\log_3 k\rceil$ is the length of the ternary representation and $n=\lceil\log_2 k\rceil$ is the length of the binary representation, $m=n\log_3 2$, *i.e.* 160-binary is equivalent to 101-ternary and 192, 256, 600-binary [21] is equivalent to 122, 162, 379-ternary, respectively. We suppose $n=160$ bits, $m=101$ bits. According to the experiment of Bernstein [3], we assume $I/M=8$, $S/M=0.8$.

Table 3 shows the comparison of Algorithm 3 and existing algorithms over GF(2^m). It can be seen that Algorithm 3 has a good improved efficiency compared with the algorithms of [12, 23] and [15]. In comparison, the improved efficiency of Algorithm 3 is 13.6%, 33.9%, 8.7%, 13.4%, 1.6%, and 15.4%, respectively.

In order to analyze the dynamic changes of the improved efficiency of Algorithm 3 than existing algorithms, we suppose

$$I/M=\beta \tag{10}$$

$S/M=0.8$. The improved efficiency of Algorithm 3, *i.e.* ε can be given as follows:

$$\varepsilon=1-\frac{(m-1)(\#I_1+\#M_1)}{\ell(\#I_2+\#M_2)} \tag{11}$$

$(\#I_1+\#M_1)$ represents the amount of calculations of Algorithm 3 per loop, and $(\#I_2+\#M_2)$ represents the amount of calculations of existing algorithms in each loop. $(m-1)$ and ℓ indicate the number of iterations of Algorithm 3 and existing algorithms.

Field inversion operations can be efficiently computed by the Extended Euclidean Algorithm (EEA) over GF(2^m), which uses $gcd(a,b)=gcd(b+ca,a)$ for all binary polynomials. According to [1], when the field size is 163 bits, performance of a field inversion operation using the EEA is equal to about 6.67-10.33 field multiplication operations in binary field, which means β is about 6.67-10.33.

Figure 1 shows the comparison of Algorithm 3 and existing algorithms. I/M is the x-axis and the improved

Table 1: The atomic block elliptic curve triple and double-and-add (AETD)

Input:$T_1 = P_1 = x_1, T_2 = P_2 = x_2, T_3 = P = x$ Output:$(3P_1, 2P_1 + P_2)$ or $(2P_1 + P_2, 2P_2 + P_1)$ or $(2P_2 + P_1, 3P_2)$		
$k_i = 0$	$k_i = 1$	$k_i = 2$
$(T_1, T_2) = (3P_1, 2P_1 + P_2)$	(T_1, T_2)=$(2P_1 + P_2, 2P_2 + P_1)$	$(T_1, T_2) = (2P_2 + P_1, 3P_2)$
$T_4 \leftarrow T_1 + T_2(x_1 + x_2)$	$T_4 \leftarrow T_1 + T_2(x_1 + x_2)$	$T_4 \leftarrow T_1 + T_2(x_1 + x_2)$
$T_5 \leftarrow {T_1}^2({x_1}^2)$	$T_5 \leftarrow {T_4}^2((x_1 + x_2)^2)$	$T_5 \leftarrow {T_4}^2((x_1 + x_2)^2)$
$T_6 \leftarrow T_3 + T_2(dummy)$	$T_6 \leftarrow T_3 + T_2(x + x_2)$	$T_6 \leftarrow T_3 + T_2(x + x_2)$
$T_6 \leftarrow T_5 \cdot T_5({x_1}^4)$	$T_6 \leftarrow T_6 \cdot T_5((x + x_2)(x_1 + x_2)^2)$	$T_6 \leftarrow T_6 \cdot T_5((x + x_2)(x_1 + x_2)^2)$
$T_7 \leftarrow T_1 \cdot T_5({x_1}^3)$	$T_7 \leftarrow T_2 \cdot T_5(x_2(x_1 + x_2)^2)$	$T_7 \leftarrow T_2 \cdot T_5(x_2(x_1 + x_2)^2)$
$T_5 \leftarrow b$	$T_4 \leftarrow b$	$T_5 \leftarrow b$
$T_8 \leftarrow T_1 \cdot T_2(x_1x_2)$	$T_8 \leftarrow T_1 \cdot T_2(x_1x_2)$	$T_8 \leftarrow T_1 \cdot T_2(x_1x_2)$
$T_5 \leftarrow T_5 + T_7(b + {x_1}^3)$	$T_4 \leftarrow T_5 + T_7(dummy)$	$T_4 \leftarrow T_5 + T_7(dummy)$
$T_5 \leftarrow T_6 + T_5(A)$	$T_4 \leftarrow T_6 + T_8(A)$	$T_4 \leftarrow T_6 + T_8(A)$
$T_6 \leftarrow T_3 + T_1(x + x_1)$	$T_6 \leftarrow T_3 + T_1(x + x_1)$	$T_6 \leftarrow T_3 + T_1(dummy)$
$T_4 \leftarrow {T_4}^2((x_1 + x_2)^2)$	$T_3 \leftarrow {T_3}^2(dummy)$	$T_6 \leftarrow {T_2}^2({x_2}^2)$
$T_9 \leftarrow T_1 \cdot T_4(x_1(x_1 + x_2)^2)$	$T_9 \leftarrow T_1 \cdot T_5(x_1(x_1 + x_2)^2)$	$T_9 \leftarrow T_2 \cdot T_6({x_2}^3)$
$T_4 \leftarrow T_6 \cdot T_4((x + x_1)(x_1 + x_2)^2)$	$T_5 \leftarrow T_6 \cdot T_5((x + x_1)(x_1 + x_2)^2)$	$T_6 \leftarrow T_6 \cdot T_6({x_2}^4)$
$T_6 \leftarrow T_6 + T_9(dummy)$	$T_6 \leftarrow T_6 + T_9(dummy)$	$T_6 \leftarrow T_6 + T_9$
$T_4 \leftarrow T_4 + T_8(B)$	$T_5 \leftarrow T_5 + T_8(B)$	$T_5 \leftarrow T_6 + T_5(B)$
$T_6 \leftarrow T_5 \cdot T_4(AB)$	$T_6 \leftarrow T_4 \cdot T_5(AB)$	$T_6 \leftarrow T_5 \cdot T_4(AB)$
$T_6 \leftarrow {T_6}^{-1}((AB)^{-1})$	$T_6 \leftarrow {T_6}^{-1}((AB)^{-1})$	$T_6 \leftarrow {T_6}^{-1}((AB)^{-1})$
$T_5 \leftarrow T_6 \cdot T_5(B^{-1})$	$T_4 \leftarrow T_6 \cdot T_4(B^{-1})$	$T_5 \leftarrow T_6 \cdot T_5(A^{-1})$
$T_4 \leftarrow T_6 \cdot T_4(A^{-1})$	$T_5 \leftarrow T_6 \cdot T_5(A^{-1})$	$T_4 \leftarrow T_6 \cdot T_4(B^{-1})$
$T_4 \leftarrow T_4 \cdot T_7(A^{-1}{x_1}^3)$	$T_5 \leftarrow T_5 \cdot T_7(A^{-1}x_2(x_1 + x_2)^2)$	$T_4 \leftarrow T_4 \cdot T_7(B^{-1}x_2(x_1 + x_2)^2)$
$T_6 \leftarrow {T_4}^2$	$T_6 \leftarrow {T_5}^2$	$T_6 \leftarrow {T_4}^2$
$T_4 \leftarrow T_4 + T_6$	$T_5 \leftarrow T_5 + T_6$	$T_4 \leftarrow T_4 + T_6$
$T_4 \leftarrow T_1 + T_4(3P_1)$	$T_5 \leftarrow T_1 + T_5(2P_2 + P_1)$	$T_4 \leftarrow T_1 + T_4(2P_2 + P_1)$
$T_5 \leftarrow T_5 \cdot T_9(B^{-1}x_1(x_1 + x_2)^2)$	$T_4 \leftarrow T_4 \cdot T_9(B^{-1}x_1(x_1 + x_2)^2)$	$T_5 \leftarrow T_5 \cdot T_9(A^{-1}{x_2}^3)$
$T_9 \leftarrow {T_5}^2$	$T_9 \leftarrow {T_4}^2$	$T_9 \leftarrow {T_5}^2$
$T_5 \leftarrow T_5 + T_9$	$T_4 \leftarrow T_4 + T_9$	$T_5 \leftarrow T_5 + T_9$
$T_2 \leftarrow T_2 + T_5(2P_1 + P_2)$	$T_1 \leftarrow T_2 + T_4(2P_1 + P_2)$	$T_2 \leftarrow T_2 + T_5(3P_2)$
$T_1 \leftarrow T_4(3P_1)$	$T_2 \leftarrow T_5(2P_2 + P_1)$	$T_1 \leftarrow T_4(2P_2 + P_1)$
$(A = {x_1}^4 + {x_1}^3 + b;$	$(A = (x + x_2)(x_1 + x_2)^2 + x_1x_2;$	$(A = (x + x_2)(x_1 + x_2)^2 + x_1x_2;$
$B = (x + x_1)(x_1 + x_2)^2 + x_1x_2)$	$B = (x + x_1)(x_1 + x_2)^2 + x_1x_2)$	$B = {x_2}^4 + {x_2}^3 + b)$

Table 2: The computation costs of Algorithm 2 under different calculation formulas

Formulas	$3P_1$	$2P_1 + P_2$	Average costs of main iteration	Anti-SPA
(2)(3)	$2I + 4M + 2S$	$2I + 4M + 2S$	$4I + 8M + 4S$	yes
(4)	$2I + 2M + 2S$	$2I + 2M + 2S$	$4I + 4M + 4S$	yes
(6)(8)	$1I + 1M + 3S + 1C$	$1I + 4M + 2S$	$2I + 6M + 14/3S + 2/3C$	no

Table 3: The computation costs of different scalar multiplication algorithms

Algorithm	Total costs of main iteration	$n = 160$bits,$m = 101$bits	Anti-SPA
Unprotected NAF [23]	$(10M + 20/3S)n$	$2453.3M$	No
Co_Z protected NAF [23]	$(85/6M + 265/36S)n$	$3208.9M$	Yes
Mont2 [12]	$3(1I + 1M + 1S)(n/2 - 1)$	$2322.6M$	No
Mont3 [12]	$37/24(1I + 1M + 1S)(n + 2)$	$2447.6M$	No
Co_Z STF [15]	$(52/3M + 5S)m$	$2154.7M$	No
Co_Z Anti-SPA STF [15]	$(20M + 6S)m$	$2504.8M$	Yes
Algorithm 3	$(1I + 10M + 4S)(m - 1)$	$2120M$	Yes

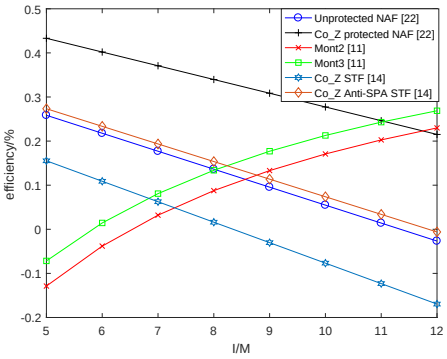


Figure 1: The comparison of Algorithm 3 and existing algorithms

efficiency of Algorithm 3 than existing algorithms is the y-axis. When $I/M = 8$, Table 3 can be obtained. It can be seen from Figure 1, for algorithms of [23] and [15], the improved efficiency of Algorithm 3, *i.e.* ε, decreases linearly as β increases. For algorithms of [12], ε increases as β increases and the larger β, the slower ε increases. When β is 6.67-10.33, Algorithm 3 is more efficient than other algorithms except for Co_Z STF algorithm [15]. However, Algorithm 3 performs better than Co_Z STF algorithm [15] when β is less than 8.3. In summary, Algorithm 3 has a good improvement in efficiency compared with existing algorithms.

5 Conclusions

In this paper, we proposed an anti-SPA scalar multiplication algorithm based on side channel atomic block over GF(2^m). Besides, we have optimized the bottom field operations by presenting new composite operation formulas $3P_1$ and $2P_1 + P_2$. Figure 1 intuitively shows the comparison of the proposed algorithm and existing algorithms. When $I/M = 8$, it can be seen from Table 3 that the proposed algorithm is more efficient than existing algorithms, ranging from 1.6% to 33.9%. Then we can apply it to the specific environments, such as wireless sensor networks with resource-limited. Next, we will try to transform the coordinate to optimize the proposed composite operation formulas and then propose a more efficient scalar multiplication algorithm.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61872058), the Key Research and Development Program of Shaanxi (Program No.2021NY-211). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] R. Avanzi and N. Thériault, "Effects of optimizations for software implementations of small binary field arithmetic," in *International Workshop on the Arithmetic of Finite Fields*, pp. 69–84, 2007.

[2] D. J. Bernstein and T. Lange, "Faster addition and doubling on elliptic curves," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 1–20, 2007.

[3] D. J. Bernstein and T. Lange, "Analysis and optimization of elliptic-curve single-scalar multiplication," *Contemporary Mathematics*, vol. 461, no. 461, pp. 1, 2008.

[4] B. Chevallier-Mames, M. Ciet, and M. Joye, "Low-cost solutions for preventing simple side-channel analysis: Side-channel atomicity," *IEEE Transactions on computers*, vol. 53, no. 6, pp. 760–768, 2004.

[5] S. M. Cho, S. C. Seo, T. H. Kim, Y. H. Park, and S. Hong, "Extended elliptic curve Montgomery ladder algorithm over binary fields with resistance to simple power analysis," *Information Sciences*, vol. 245, pp. 304–312, 2013.

[6] L. Elmegaard-Fessel, "Efficient scalar multiplication and security against power analysis in cryptosystems based on the NIST elliptic curves over prime fields," *IACR Cryptology ePrint Archive*, vol. 2006, pp. 313, 2006.

[7] R. R. Farashahi, H. F. Wu, and C. A. Zhao, "Efficient arithmetic on elliptic curves over fields of characteristic three," in *International Conference on Selected Areas in Cryptography*, pp. 135–148, 2012.

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

[9] N. KOBLITZ, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[10] Z. X. Lai and Z. J. Zhang, "Scalar multiplication on hessian curves based on Co_Z operations," *Bulletin of Science and Technology (in Chinese)*, vol. 32, no. 2, pp. 28–33, 2016.

[11] L. Li, "Research on the ternary algorithm in the elliptic curve operations," *Journal of Network Safety Technology and Application (in Chinese)*, no. 11, pp. 94–96, 2015.

[12] Y. Li, J. L. Wang, X. W. Zeng, and X. Z. Ye, "A segmented Montgomery scalar multiplication algorithm with resistance to simple power analysis," *Computer Engineering and Science (in Chinese)*, vol. 39, no. 1, pp. 92–102, 2017.

[13] H. Z. Liu, Q. H. Dong, and Y. B. Li, "Efficient ECC scalar multiplication algorithm based on symmetric ternary in wireless sensor networks," in *Progress in Electromagnetics Research Symposium-Fall*, pp. 879–885, 2017.

[14] S. G. Liu, R. R. Wang, Y. Q. Li, and C. L. Zhai, "An improved ternary Montgomery ladder algorithm on

elliptic curves over GF(3ˆm)," *International Journal Network Security*, vol. 21, no. 3, pp. 384–391, 2019.

[15] S. G. Liu, Y. Y. Ding, R. Shi, and S. M. Lu, "Co_Z addition on elliptic curves over finite fields GF(2ˆm)," *Journal of Wuhan University (in Chinese)*, vol. 65, no. 2, pp. 207–212, 2019.

[16] J. López and R. Dahab, "Fast multiplication on elliptic curves over GF(2ˆ m) without precomputation," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 316–327, 1999.

[17] V. S. Miller, "Use of elliptic curves in cryptography," in *Conference on the Theory and Application of Cryptographic Techniques*, pp. 417–426, 1985.

[18] P. L. Montgomery, "Speeding the Pollard and elliptic curve methods of factorization," *Mathematics of Computation*, vol. 48, no. 177, pp. 243–264, 1987.

[19] J. Moon, D. Lee, and J. Jung, "Improvement of efficient and secure smart card based password authentication scheme," *International Journal of Network Security*, vol. 19, no. 6, pp. 1053–1061, 2017.

[20] T. Oliveira, J. López, and F. Rodríguez-Henríquez, "The Montgomery ladder on binary elliptic curves," *Journal of Cryptographic Engineering*, vol. 8, no. 3, pp. 241–258, 2018.

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

[22] H. Wang, B. Li, and W. Yu, "Montgomery algorithm on elliptic curves over finite fields of character three," *Journal on Communications (in Chinese)*, vol. 29, no. 10, pp. 25–29, 2008.

[23] J. Wei, X. Liu, H. Liu, and W. Guo, "A low-time-complexity and secure dual-field scalar multiplication based on co-z protected NAF," *IEICE Electronics Express*, vol. 11, no. 11, pp. 20140361–20140361, 2014.

[24] X. S. Yan, X. G. Zhang, H. F. Zhang, and L. Liu, "Countermeasures in CPU for timing and power side channel attack," *DEStech Transactions on Engineering and Technology Research*, 2018. DOI: 10.12783/dtetr/pmsms2018/24880.

[25] J. You, Q. Zhang, C. D'Alves, B. O'Farrell, and C. K. Anand, "Using z14 fused-multiply-add instructions to accelerate elliptic curve cryptography," in *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, pp. 284–291, 2019.

[26] W. Yu, B. Li, K. W. Wang, and W. H. Li, "Co_Z Montgomery algorithm on elliptic cureves over finite fields of characteristic three," *Journal of Computer (in Chinese)*, vol. 40, no. 5, pp. 1121–1131, 2017.

[27] W. Yu, S. A. Musa, and B. Li, "Double-base chains for scalar multiplications on elliptic curves," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 538–565, 2020.

[28] N. Zhang and Q. Q. Peiand G. Z. Xiao, "Elliptic curve scalar multiplication with x-coordinate," *Wuhan University Journal of Natural Sciences*, vol. 12, no. 1, pp. 163–166, 2007.

Biography

Shuang-Gen Liu, born in 1979. He received the PH.D. in cryptography form Xidian University in 2008. He is currently an associate professor with the school of cyber security, Xian University of Posts and Telecommunications, Xi?an, China. He is a member of the China Computer Federation, and a member of the Chinese Association for Cryptologic Research. His recent research interests include crptography and information security.

Yan-Yan Hu, born in 1995. A graduate student of Xi'an University of posts and telecommunications. She is mainly engaged in the research of elliptic curve cryptosystem.

Lan Wei, born in 1998. An undergraduate student of Xi'an University of posts and telecommunications.She is mainly engaged in the research of elliptic curve cryptosystem.

Research on Dynamic Social Network Anonymity Technology for Protecting Community Structure

Na Li[1], Xiao-Lin Zhang[1], Yong-Ping Wang[1], Jian Li[1], and Li-Xin Liu[2]
(Corresponding author: Xiao-Lin Zhang)

School of Information Engineering, Inner Mongolia University of Science and Technology[1]
Baotou 014010, China
School of Information, Renmin University of China[2]
Beijing 100872, China
Email: zhangxl@imust.edu.cn
(Received Feb. 8, 2020; Revised and Accepted Aug. 23, 2020; First Online May 30, 2021)

Abstract

The dynamic change of vertex degree in a dynamic social network will lead to vertex identity disclosure given the deficiencies in current privacy protection methods, such as the destruction of community structure and low data processing capability of a single workstation. The dynamic social network degree sequence anonymity (DSNDSA) method to protect community structure is proposed. The method obtains the grouping and anonymous results based on a compressed binary tree constructed by a new method called a multidimensional vector. Dummy vertices are added in parallel to construct anonymous graphs. Distributed to merge dummy vertices method based on the community is designed to reduce the number of vertices added to satisfy the anonymity model. A divide and the agglomerate algorithm is expanded for community detection. The experimental results show that the proposed algorithm based on GraphX can overcome the defects of the traditional algorithm in community protection while meeting the requirement of anonymity.

Keywords: Compressed Binary Tree; Community Structure; Divide and Agglomerate; Dynamic Social Network Anonymity; GraphX

1 Introduction

With the advent of web 2.0, social networking platforms are becoming more and more popular,such as Facebook, Twitter, LinkedIn and Google+ [1]. These social networking platforms has generated a large amount of data.Massive social network data attracts vast researchers with its great research value.Because these data carries the user's privacy informations inevitably, it is imperative for the data owner to protect the privacy of the information before releasing the data to the third-party.

In order to resist attacks of various background knowledge on users′privacy, a variety of social network anonymization technologies have emerged in recent years. These technologies are mainly focused on single graph anonymity [12]. Due to the existence of a large number of incremental social networks in real life, attackers are likely to re-identify the target vertex with the vertex degree information on two consecutive timestamps as background knowledge. Therefore, it has practical significance to abstract the dynamic social network graph into a set of simple incremental sequences and protect its privacy [7]. In addition, many real-world networks are organized according to a community structure intimately. Much research effort has been devoted to develop methods and algorithms that can efficiently highlight this hidden structure of a network [6]. However, the existing dynamic social network privacy protection models still has great limitation on community structure protection. This has affected that researchers attempts to analyze the characteristics of social networks according to the network community structure, which has reduced the availability of data greatly.

To solve the above problems, a distributed anonymity algorithm of dynamic social network is designed to protect the community structure. The main contributions of this paper are as follows:

1) Aiming at the undirected graph of dynamic social network, a degree sequence attack model is defined, and a dynamic social network degree sequence anonymity model for protecting community structure is proposed, thus preventing privacy attacks by attackers with vertex degree sequences as background knowledge effectively.

2) A privacy protection algorithm of dynamic social network k-degree sequence is designed to protect the community structure while ensuring that anonymous graphs satisfies k-degree sequence anonymity on different timestamps.

3) Experimental tests and analysis on real data sets verifies that the validity of dynamic social network degree sequence anonymity model and the high availability of dynamic social network anonymous graphs in the aspect of community structure.

The rest of this paper is organized as follows. Section 2 reviews the previous related research in more detail. Section 3 formalizes related definition and privacy model. Section 4 presents the DSNDSA algorithm and we conduct experiments on real data sets in Section 5. Finally, the conclusion of the paper is given in Section 6

2 Related Work

Recent development in the technology has made it easier to collect massive amounts of social network data and leads to serious privacy problems. Regarding the privacy information to be preserved in social networks, three main categories of privacy threats have been identified [5]: identity disclosure [4, 7, 8, 10, 14, 15, 21, 22], attribute disclosure [3] and link disclosure [16, 18, 20]. Due to the diversity of privacy threats, more and more researchers has focused their attention on the protection of social network privacy and proposed a variety of anonymity methods.K-anonymity framework is a classic anonymity framework for social networks.K-anonymity requires that for any element in a set, there are at least k-1 duplicate elements that are the same as it, *i.e.* any element can be identified with a probability no greater than 1/k in a set. This paper mainly studies the identity disclosure of social network vertex based on k-anonymity framework.

Identity disclosure includes sub-categories such as vertex existence, vertex properties and graph metrics [5]. Literature [10] introduces a time-saving k-degree anonymization method TSRAM in social network that without having to rescan the data set for different levels of anonymity. It ensures that the attacker takes vertex degree as background knowledge and the probability of successfully identifying the target vertex does not exceed 1/k. Literature [21] designs a general k-anonymization framework, which can be used with various utility measurements to achieve k-anonymity with small utility loss on given social networks. In this method, utility measurements are designed based on more complex community-based graph models includes flat community-based utility model and Hierarchical community-based utility model. Literature [22] addresses the problem of excessive loss of graphlet structural information in the privacy process of published social network data, and proposed a technique of hierarchical k-anonymity for graphlet structural perception. The mehod divide the degrees of nodes according to the degree to which the social networking graphical node obeys the characteristics of a power-law distribution, and the divided nodes define the different privacy levels according to their practical means. The purpose is meet the privacy requirement while protecting the graphical structural information in the social network and improving the utility of the data. Literature [14] considers protecting the weighted social networks from weight-based attacks and propose a method KWGA based on the weighted social networks. And This method combines k-anonymous with generalization method to ensure the security of the social network data when it is published. Literature [15] proposes an improved k-degree anonymity model that provides privacy with low utility loss. This method performs anonymous operations on the basis of dividing communities, making the vertices of the same group indistinguishable based on the degree value.

Social network data publishing is dynamic. Literature [7] proposes a weighted graph incremental sequence k-anonymous privacy protection model, and design a baseline anonymity algorithm WLKA based on weight list and HVKA algorithm based on hypergraph, which prevents the attacks from node point labels and weight packages. Literature [8] makes one-hop neighbor's network structure and label as attacker' background knowledge and define the label neighborhood attack model in dynamic social network releases. A dynamic-l-diversity anonymized method is proposed to resist attacks. And ensuring each vertex with a sensitive label, which can't be identified in the social network with a probability higher than 1/ l. Literature [18] abstracts social networks on different timestamps into a set of simple incremental sequences, and proposes a social network anonymity method DMRA for simultaneous publication of multiple social network graphs. The method ensures that the attacker can successfully infer that the probability of a user participating in any edge and the probability of edge connection between any two vertices are not more than 1/k without any background knowledge. Literature [9] proposes a dynamic k^w-Number of Mutual Friend anonymity algorithm for protecting edge identities of dynamic networks that is released sequentially. The k^w-NMF algorithm anonymizes each release of network data so that the adversary can not re-identify the victim by knowing the knowledge of each release. Literature [17] proposes a new privacy model, dynamic k^w-structural diversity anonymity, for protecting the vertex and multicommunity identities in sequential releases of a dynamic network.

In recent years, the discovery and analysis of community structures in social networks play an important role in studying the characteristics of complex networks. Correspondingly, the combination of social network anonymization and protection of community structure has attracted the attention of many scholars. Literature [11] uses the concept of upper approximation of the original set to propose a social network privacy protection method PPGP. The method can effectively protect the graphic community structure in the anonymous process. And it makes the anonymous social network graph have good performance in graph mining tasks such as clustering, classification and PageRank computation. Literature [19] proposes a novel local perturbation technique that can reach the same privacy requirement of k-anonymity, while minimizing the impact on commu-

nity structure. Literature [23] proposes a probabilistic anonymizing method to protect the data privacy, which combines k-anonymous with random perturbation. The proposed method can minimize the impact on community structure.

To sum up, different social network anonymity methods can resist different privacy attacks. Most privacy protection models are aimed at a single social network graph, and single graph anonymity technology is not sufficient to cope with the dynamic changes of social networks. In addition, the existing dynamic social network privacy protection technologies ensures the availability of the original social network data while achieving anonymity. But they ignored the protection of the social network community structure. Therefore, this paper studies the vertex identity re-identification of dynamic social networks, taking the degree sequence of a vertex on different timestamps as the attacker's background knowledge. And a dynamic social network anonymity algorithm DSNDSA is proposed, which can protect the social network community structure in the process of social network anonymity effectively.

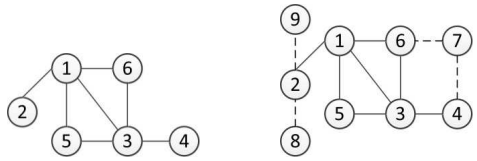
3 Definitions and Dynamic Social Network Degree Sequence Anonymity Model

Definition 1. *(Incremental dynamic social network graph) The sequence of social network graphs on different timestamps is denoted* $g = \langle G_0, G_1, ..., G_t \rangle$*, The social network graph at time t is denoted* $G_t = (V_t, E_t)$*, where* V_t *is a set of vetices representing users at time t,* E_t *is a set of edges representing the interaction among users at time t. With the passage of time, we assume t hat the vertices and edges of social network graph are not decreasing,* i.e. $V_{t-1} \subseteq V_t, E_{t-1} \subseteq E_t$*. The sequence g of social network graph like this is called incremental dynamic social network graph. Figure 1(a) and Figure 1(b) are incremental dynamic social network graphs on two consecutive timestamps.*

Definition 2. *(Degree sequence of vertex) Given a incremental dynamic social network graph* $g = \langle G_i, G_{i+1}, \cdots, G_t \rangle$*. if* $\forall v \in V_t$*, the degree of vertex v in social network graph* G_i *is denoted* $d_{(v,G_i)}$*.*$\Delta_v = (d_{(v,G_i)}, d_{(v,G_{i+1})}, ..., d_{(v,G_t)})$ *is called degree sequence of vertex v in incremental dynamic social network graph g.*

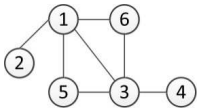
Definition 3. *(First category vertex and second category vertex) Given a incremental dynamic social network graph* $g = \langle G_{t-1}, G_t \rangle$*.*$V_g = \{v_1, v_2, ..., v_i, ..., v_n\}$ *is a set of all vertices in g.*v_i *belongs to the first category vertex if* $v_i \in V_{t-1}$ *and* $v_i \in V_t$*. Otherwise,*v_i *belongs to the second category vertex if* $v_i \notin V_{t-1}$ *and* $v_i \in V_t$*.*

Definition 4. *(Multidimensional vector) Given a incremental dynamic social network graph* $g = \langle G_{t-1}, G_t \rangle$ *(*$v \in V_t$*). One of the Multidimensional vector corresponding to g is denoted* (Δ_v, C)*, where* Δ_v *is degree sequence of vertex v in incremental dynamic social network graph g, C is the count of vertices whose degree sequence equal to* Δ_v*.*

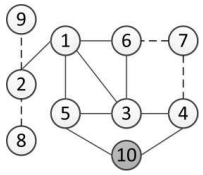


(a) Original graph G_{t-1} at time t-1

(b) Original graph G_t at time t



(c) Anonymous graph G^*_{t-1} at time t-1



(d) Anonymous graph G^*_t at time t

Figure 1: Incremental dynamic social network original graph and anonymous graph

Definition 5. *(Dynamic social network vertex degree sequence attack) Given a incremental dynamic social network graph* $g = \langle G_{t-1}, G_t \rangle$ *on two consecutive timestamps. An attacker can successfully identify the target vertex v with the degree sequence* Δ_v *of vertex v as background knowledge, which is called dynamic social network vertex degree sequence attack.*

As shown in Figure 1(a) and Figure 1(b), it is assumed that the attacker knows that the degree of the target vertex Bob in G_{t-1} and G_t is 1 and 2 respectively. Since the release graphs at both timestamps satisfy 2- degree anonymity, the probability of the attacker identifying Bob correctly is 1/2 at each timestamp. If the attacker combines the anonymous graphs on two timestamps, *i.e.* Δ_v=(1,2) as background knowledge, he can identify that the vertex 4 is Bob with 100% probability successfully. In the following, according to the dynamic social network vertex degree sequence attack model, the dynamic social network vertex degree sequence k-anonymity is defined.

Definition 6. *(Vertex k-degree sequence anonymity) Given a incremental dynamic social network original graph* $g = \langle G_{t-1}, G_t \rangle$ *on two consecutive timestamps and the privacy parameter k. Incremental dynamic social network anonymous graph is denoted* $g^* = \langle G^*_{t-1}, G^*_t \rangle$*. For the degree sequence* Δ_v *of any one first category vertex v(second category vertex), there are at least k-1 other first category vertices(second category vertices) with the same degree sequence in* g^**. The degree of privacy protection increases with the increase of k. It is said that the* g^* *satisfies vertex k-degree sequence anonymity.*

As shown in Figure 1(a) and Figure 1(b), in order to make the dynamic social network meets the anonymity

requirement of vertex k-degree sequence, a dummy vertex 10 and two dummy edges e(4, 10) and e(5, 10) are added to G_t. The 2-degree sequence anonymity graphs are shown in Figure 1(c) and Figure 1(d). For any vertex v, there is at least one vertex with the same degree sequence as vertex v, that is, the attacker cannot uniquely identify the target vertex with a probability greater than 1/2. Anonymous graphs satisfies vertex 2-degree sequence anonymity.

Definition 7. *(Dynamic social network degree sequence anonymity model) Given a incremental dynamic social network original graph* $g = \langle G_{t-1}, G_t \rangle$ *on two consecutive timestamps and a positive integer k which can adjust the degree of anonymity. If the Incremental dynamic social network anonymous graph* $g^* = \langle G^*_{t-1}, G^*_t \rangle$ *meets the following four requirements, It is said that the* g^* *conforms to the dynamic social network degree sequence anonymity model for protecting community structure.*

1) For the dynamic social network anonymous graph G^*_t at any timestamp, the probability that the attacker identifies the target vertex successfully based on the vertex degree does not exceed 1/k;

2) For the incremental dynamic social network anonymous graph $g^* = \langle G^*_{t-1}, G^*_t \rangle$, the probability that the attacker identifies the target vertex successfully based on the degree sequence of vertex does not exceed 1/k;

3) In the process of anonymity, original vertices and original edges does not change;

4) Social network community structure is protected in the process of social network anonymity.

4 Dynamic Social Network Degree Sequence Anonymity (DSNDSA) Algorithm for Protecting Community Structure

The solution to anonymize dynamic social network graph g is detailed in this section. The algorithm DSNDSA includes three steps:

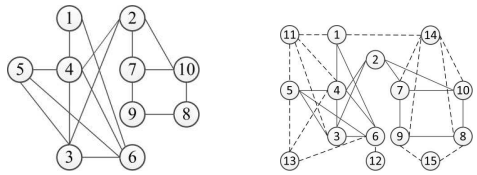
1) Community detection;

2) Vertex grouping and anonymity;

3) Graph reconstruction.

4.1 Community Detection

Given a incremental dynamic social network graph g = $\langle G_{t-1}, G_t \rangle$, the type and number of social network communities is unchanged with the passage of time. The social network vertices in G_{t-1} is divided into different communities by DA algorithm [13]. Each vertex added at timestamp t is regarded as a sub-graph that does not meet the community criterion, and the community to which the added vertex belongs is selected through the biggest AT index for the community to attract sub-graph [13]. Similarly, dummy vertices use the same community detection method.



(a) Original graph G_{t-1} at time t-1 (b) Original graph G_t at time t

Figure 2: Dynamic social network graph

Figure 2 is a dynamic social network original graph g = $\langle G_{t-1}, G_t \rangle$. The 10 vertices in G_{t-1} are divided into two communities using the DA algorithm, C_1={1,3,4,5,6} and C_2={2,7,8,9,10}. Taking the vertex 11 which is newly added in G_t as an example, the AT indexes of two communities to the vertex 11 are calculated respectively, *i.e.* $AT_{11,C_1} = 3/2, AT_{11,C_2} = 0$. Therefore, vertex 11 belongs to community C_1. Updating community collections C_1={1,3,4,5,6,11,12,13} and C_2={2,7,8,9,10, 14,15}.

4.2 Vertex Grouping and Anonymity

Vertex grouping and anonymity is to group vertices of dynamic social network with the objective of minimizing anonymity cost and to determine the anonymity degree sequence of vertices in the grouping. In order to achieve this goal, different grouping and anonymity methods are proposed for different categories of dynamic social network vertices.

4.2.1 Grouping and Anonymity for First Category Vertex

Given a incremental dynamic social network graph g = $\langle G_{t-1}, G_t \rangle$, all vertices in G_{t-1} belongs to the first category vertex. According to literature [10], the process of grouping and anonymity for the first category vertex includes the following three steps:

1) Generating Multidimensional Vector: According to the dynamic social network graph, the vertex degree sequences of the first category vertices are obtained, and one or more vertices with the same vertex degree sequence are expressed as multidimensional vectors.

2) Constructing Compressed Binary Tree: The multidimensional vectors are sorted to generate leaf nodes of a binary tree, and merging leaf nodes and calculating multidimensional vectors of parent nodes based on the "travel time" criterion. This process is iterated many times until the binary tree is constructed. The

compressed binary tree can reflect degree sequence of the first category vertices in g.

3) Cutting Line Drawing on the Tree: Cut a line on the tree according to the privacy parameter k to obtain the grouping result and anonymity degree sequence of the first category vertex. The higher the degree of anonymity, the closer the tangent is to the root.

In the above steps, the leaf nodes of binary tree are represented by multidimensional vectors composed of vertex degree sequences and counts.Since the social network is incremental, the dimensions of the multidimensional vectors corresponding to the first category vertices are consistent. In order to minimize the anonymity cost of the first category vertices anonymity, this paper sorts multidimensional vectors from left to right according to the following rules:

1) Given the degree sequence of the first category vertices $\Delta_v = (d_{(v,G_{t-1})}, d_{(v,G_t)})$, Sorting multidimensional vectors in descending order according to $\sum_{i=t-1}^{t} d_{(v,G_i)}$;

2) If $\sum_{i=t-1}^{t} d_{(v,G_i)}$ are same between them, Sorting in descending order according to the value of the first element in Δ_v, and so on.

Finally, the parent node generated by merging the two leaf nodes Lm and Ln is represented by multidimensional vector $(\max\{d_{(L_m,G_{t-1})}, d_{(L_n,G_{t-1})}\}, \max\{d_{(L_m,G_t)}, d_{(L_n,G_t)}\}, C_m + C_n)$. In the similarity calculation process, the potential fields are calculated for all the data based on Euclidean distance between multidimensional vectors, *i.e.* Euclidean distance in three-dimensional space corresponding to the first category vertex.

As shown in Figure 2, the set of first category vertices is {1,2,3,4,5,6,7,8,9,10}. Firstly, ten first category vertices are represented by seven multidimensional vectors, as shown in Table 1.

Table 1: Multidimensional vector corresponding to first category vertices in dynamic social network

Vector ID	Degree sequence	Count	Total degree	Multidimensional vector
1	(5,7)	1	12	(5,7,1)
2	(4,4)	1	8	(4,4,1)
3	(3,5)	1	8	(3,5,1)
4	(4,5)	1	9	(4,5,1)
5	(4,6)	1	10	(4,6,1)
6	(2,4)	3	6	(2,4,3)
7	(3,4)	2	7	(3,4,2)

Secondly, the purpose of calculating the similarity between multidimensional vectors is merging leaf nodes. For example, $S_{45} = \frac{\phi_4 - \phi_5}{r_{45}^2} = 0.7467602106$ and $S_{42} = \frac{\phi_4 - \phi_2}{r_{24}^2} = 0.773902479$. Because $S_{42} > S_{45}$,parent node (4,5,2) is generated by merging leaf node (4,5,1) and leaf node (4,4,1). Finally, the structure and cut line position of the compressed binary tree corresponding to the first category vertices is shown in Figure 3 when the anonymity parameter k is 3. Therefore, grouping results of the first category vertices is $Group_1$={2,3,4,6},$Group_2$={5,7,10} and $Group_3$={1,8,9}. The anonymity degree sequences corresponding to them are $\Delta_v^*(Group_1) = (5,7)$,$\Delta_v^*(Group_2) = (3,5)$ and $\Delta_v^*(Group_3) = (2,4)$.

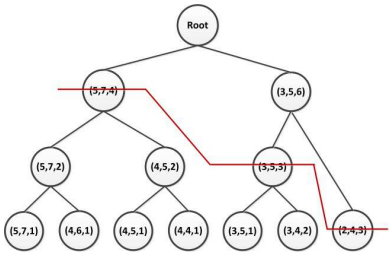


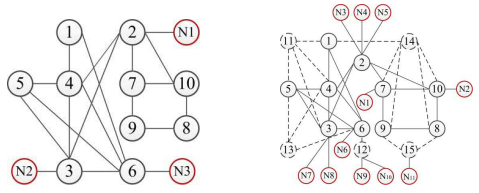
Figure 3: Compressed binary tree corresponding to the first category vertex

4.2.2 Grouping and Anonymity for Second Category Vertex

The essence of the second category vertex anonymity is single graph anonymity. The process of grouping and anonymity is divided into the following two steps:

1) Remove the second category vertices whose degree is in the set DSet(t) or who satisfies the k-anonymity condition themselves, and the remaining vertices are called the sequence of vertices who will be anonymized. And the elements in set DSet(t) are composed of the anonymious degrees of the first category vertices in G_t.

2) If the count of second category vertices who will be anonymized is not less than k, the anonymity process of it is similar to the first category vertex, where leaf nodes are represented by two-dimensional vectors. On the contrary, the anonymous degree with the smallest difference compared to its degree and greater than its own degree is selected as its target degree for every second category vertex who will be anonymized.



(a) G'_{t-1} (b) G'_t

Figure 4: Dynamic social network initial anonymous graph $\langle G'_{t-1}, G'_t \rangle$

As shown in Figure 2, the set of second category vertices is {11,12,13,14,15} and DSet(t)={4,5,7}. Therefore, the set of second category vertices who will be anonymized is {12,13,15}. The corresponding anonymious degrees are 3, 3 and 3 when k=3 separately.

The algorithm DSNDSA adds dummy vertices to the original dynamic social network in parallel based on GraphX to achieve the requirement of vertex k-degree sequence anonymity. The initial anonymious graph of dynamic social network is shown in Figure 4.

4.3 Graph Reconstruction

In order to reduce the number of dummy vertices in the social network publishing graphs and improve the usability of community structure, DSNDSA algorithm designs dummy vertex removal-addition conditions and rules based on the community to which the vertex belongs. Priority is given to dummy vertex removal-addition operations in the same community, and then dummy vertex removal-addition operation between different communities is performed.

The graph reconstruction process is implemented based on the distributed graph processing system spark graphX, which follows the characteristic of "node-centered" and improves the efficiency of privacy protection technology in processing large-scale graph data by means of message transmission between vertices. In order to select a suitable dummy neighbor vertex, n-hop dummy neighbor table(NDT) information is needed. The vertex data structure is represented by quintuple(NID,deg(N_u),com, type, tag), and each quintuple is a dummy neighbor table entry(DNTE).

1) NID:the vertex ID;

2) Deg(N_u):the degree of vertex u;

3) Com: community to which vertex u belongs;

4) Type:type of vertex u. Type=0 means that vertex u belongs to dummy vertex, type=1 and type=2 means that vertex u belongs to the first category vertex and the second category vertex respectively;

5) Tag: whether degree deg(N_u) of vertex u exists in anonymous degree set DSet(i), if so, tag=1, otherwise, tag=0. Therefore, all vertices tag=0 before adding dummy vertices.

Definition 8. *(Removal-addition condition in same community, RACSC) If dummy vertex N_u and N_v can remove-add in the same community, then N_u and N_v must meet the following three conditions at the same time:*

1) $deg(N_u)+deg(N_v) \leqslant SC\_deg_{max}$, where SC_deg_{max} represents the maximum anonymious degree of vertices in the community to which N_u belongs;

2) For the vertex N_w with type$\neq$0, the edge $e(N_u,N_w)$ and the edge $e(N_v,N_w)$ do not exist at the same time;

3) $N_u.com$=$N_u.com$.

Definition 9. *(Removal-addition rule in same community, RARSC) For any vertex N_u with type=0, the DNT of N_u is obtained through message transfer mechanism. For any vertex N_v in DNT is placed in the candidate set $N_u.CandiSet\_sc$ of N_u if it meets RACSC. The removal-addition rules in same community are as follow:*

1) If the element in the set $N_u.CandiSet\_sc$ is unique, it is the best candidate vertex of N_u;

2) If the number of elements in the set $N_u.CandiSet\_sc$ is greater than 1, the dummy vertex N_v with small value of $deg(N_u)+deg(N_v)$ is considered for remove-adding preferentially.

Any dummy vertex N_u selects the best dummy vertex algorithm SCS as follows:

Algorithm 1 Same Community Select (SCS)

Input: N_u.CandiSet_sc
Output: N_v

```
1:  if (Nu.CandiSet_sc.size > 1) then
2:    for (each N_v in N_u.CandiSet_sc) do
3:      deg(N_r)=deg(N_u)+deg(N_v);
4:    end for
5:    M=the number of dummy nodes with degree equal
      to min(deg(N_r))
6:    if (M = 1) then
7:      return N_v
8:    else
9:      randomly select N_v
10:     return N_v
11:   end if
12: else
13:   return N_v
14: end if
```

Algorithm 2 shows dummy vertex removal-addition algorithm SCRA in the same community. Lines 3-22 remove-adds dummy vertices in parallel after all supersteps are completed. Among them, lines 2-4 looks for dummy vertex neighbor information based on Pregel model. Lines 5-9 will form a candidate set of dummy vertices N_u with the virtual neighbor information that meets the RACSC. Lines 10-18 remove-adds dummy vertices N_u and N_v.

In each superstep before parallel removal-addition of dummy vertices, DSNDSA algorithm ensures that each dummy vertex is remove-added at most once by traversing the dummy vertices and updating the candidate set of dummy vertices continuously. Taking the initial anonymous graph G'_t at time t in Figure **??** as an example, the dummy vertex receives its own 3-hop dummy vertex neighbor information, thus obtaining the dummy vertex candidate set as shown in Table 2 when superstep=3. Virtual vertex N_1 selects N_2 as the best candidate vertex according to algorithm 1, and the candidate set information

updated for the first time is shown in column 3 of Table 2. By analogy, the result of selecting the best candidate vertex for all dummy vertices is shown in column 5 of Table 2. The state of virtual vertices N_1, N_2, N_6 and N_7 is set to Inactive. This means that they will not participate in the other supersteps after superstep=3. And the dummy vertex removal-addition operation is performed after all supersteps are completed. The algorithm iterates twice, and the result of remove-adding dummy vertices in the same community is shown in Figure 5.

Algorithm 2 Same Community Remove_Add (SCRA)

Input: G'
Output: $G^{\#}$

for (SuperStep=1 to 6) **do**
  sendMessToNeighbors
  for (each dummy vertex N_u in G'_i) **do**
    update N_u.DNT
    for (each N_v in N_u.DNT) **do**
      if (N_v satify RACSC) **then**
        N_u.CandiSet_sc ← N_v
      end if
    end for
    if (N_u.CandiSet_sc ⩾ 1) **then**
      N_v=Same Community Select(N_u.CandiSet_sc)
      G′.EdgeRDD.Remove ⟨ m,N_u ⟩
      G′.EdgeRDD.Remove ⟨ n,N_v ⟩
      G′.EdgeRDD.Add ⟨ m,N_r ⟩
      G′.EdgeRDD.Add ⟨ n,N_r ⟩
      NDRDD.Add (N_u,N_v)
      VoteToHalt (N_u,N_v)
    end if
  end for
end for
return $G^{\#}$

Table 2: Dummy vertex candidate set update table when surperstep=3

Dummy vertex number	candidate set	first update	second update	best candidate vertex
N_1	N_2,N_3,N_4,N_5	N_2	N_2	N_2
N_2	N_1,N_3,N_4,N_5	N_1	N_1	N_1
N_3	N_1,N_2	∅	∅	
N_4	N_1,N_2	∅	∅	
N_5	N_1,N_2	∅	∅	
N_6	N_7,N_8,N_9,N_{10}	N_7,N_8,N_9,N_{10}	N_7	N_7
N_7	N_6	N_6	N_6	N_6
N_8	N_6	N_6	∅	
N_9	N_6	N_6	∅	
N_{10}	N_6	N_6	∅	
N_{11}	∅	∅	∅	

Definition 10. *(Removal-addition condition in different communities, RACDC) If dummy vertex N_u and N_v can remove-add in the different communities, then N_u and N_v must meet the following three conditions at the same time:*

1) $deg(N_u)$+$deg(N_v)$ ⩽ deg_{max} and $deg(N_u)$+$deg(N_v)$ ∈ DSet(i), where deg_{max} represents the maximum anonymious degree in current social network;

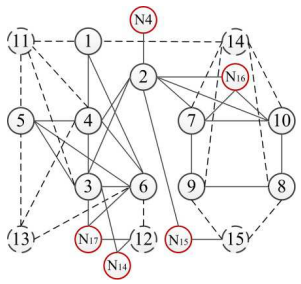


Figure 5: Result of remove-adding dummy vertices in the same community

2) For the vertex N_w of type ≠ 0, the edge $e(N_u,N_w)$ and the edge $e(N_v,N_w)$ do not exist at the same time;

3) $N_u.tag \wedge N_v.Tag = 0$.

Algorithm 3 Different Community Select(DCS)

Input: N_u.CandiSet_dc
Output: N_v

if (N_u.CandiSet_dc.size > 1) **then**
  for (each N_v in N_u.CandiSet_dc) **do**
    if (N_u.tag=0 && N_v.tag=0) **then**
      $List1_u$ ← N_v
    else
      $List2_u$ ← N_v
    end if
  end for
  if ($List1_u$.size!=0) **then**
    List= $List1_u$
  else
    List= $List2_u$
  end if
  for (each N_v in List) **do**
    deg(N_r)=deg(N_u)+deg(N_v)
  end for
  M=the number of dummy nodes with degree equal to min(deg(N_r))
  if (M = 1) **then**
    return N_v
  else
    randomly select N_v
    return N_v
  end if
else
  return N_v
end if

Definition 11. *(Removal-addition rule in different communities, RARDC) For any vertex N_u with tag=0, the DNT of N_u is obtained. For any vertex N_v in DNT is placed in the candidate set N_u.CandiSet_dc of N_u if it meets RACDC. The removal-addition rules in different communities are as follow:*

1) If the element in the set N_u.CandiSet_dc is unique, it is the best candidate vertex of N_u;

2) If the number of elements in the set N_u.CandiSet_dc is greater than 1, dummy vertex N_v with tag =0 is selected for remove-adding preferentially, and then select dummy vertex N_v who has minimum value of deg(N_u)+deg(N_v).

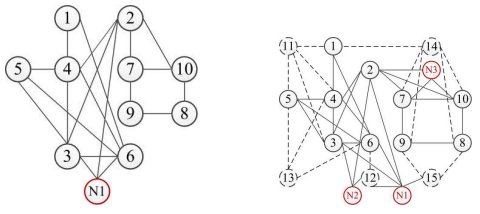
Any dummy vertex N_u selects the best dummy vertex algorithm in different communities DCS as show in Algorithm 3.

If there is dummy vertex N_u in the social network with tag=0 after algorithm 2 is executed, the dummy vertex removal-addition in different communities algorithm as shown in algorithm 4 is executed once. Lines 2-9 obtains candidate vertex sets of dummy vertices with tag=0. And lines 10-18 remove-adds dummy vertices to obtain social network publishing graph G_i^*.

Algorithm 4 Different Community Remove_Add(DCRA)

Input: $G^{\#}$
Output: G^*

1: **for** (SuperStep = 1 to 6) **do**
2: sendMessToNeighbors
3: **if** (dummy node N_u.tag=0 in $G_i^{\#}$) **then**
4: update N_u.DNT
5: **for** (each N_v in N_u.DNT) **do**
6: **if** (N_v satify RACDC) **then**
7: N_u.CandiSet_dc $\leftarrow N_v$
8: **end if**
9: **end for**
10: **if** (N_u.CandiSet_dc $\geqslant$ 1) **then**
11: N_v=Different Community Select(N_u.CandiSet_dc)
12: $G^{\#}$.EdgeRDD.Remove $\langle$ m,N_u $\rangle$
13: $G^{\#}$.EdgeRDD.Remove $\langle$ n,N_v $\rangle$
14: $G^{\#}$.EdgeRDD.Add $\langle$ m,N_r $\rangle$
15: $G^{\#}$.EdgeRDD.Add $\langle$ n,N_r $\rangle$
16: NDRDD.Add (N_u,N_v)
17: VoteToHalt (N_u,N_v)
18: **end if**
19: **end if**
20: **end for**
21: return G^*



(a) G_{t-1}^* (b) G_t^*

Figure 6: Dynamic social network release graph

In Figure 5, dummy vertices N_4, N_{14}, and N_{15} does not satisfy anonymity requirement. And the social network release graph G_t^* is obtained by executing the DCRA algorithm as shown in Figure 6(b). All vertices meets the anonymity requirement of k=3.

DSNDSA algorithm is shown in Algorithm 5. The corresponding dynamic social network release graph of Figure 2 is shown in Figure 6.

Algorithm 5 DSNDSA

Input: $g = \langle G_{t-1}, G_t \rangle$,k
Output: g^*=$\langle G_{t-1}^*, G_t^* \rangle$

1: Obtaining social network anonymious degree sequences on different timestamps
2: **for** (i=t-1 to t) **do**
3: Adding dummy vertices to original graph G_i according to anonymious degree sequence to generate initial anonymous graph G_t'
4: Initial: G_i',DNRDD=$\varnothing$, N_u.CandiSet_sc=$\varnothing$, N_u.CandiSet_dc=$\varnothing$
5: EdgeRDD=G_i'.EdgeRDD
6: update Det(i)
7: **while** ($\exists N_u, N_v \in G_i'$ && N_u, N_v satisfy RACSC) **do**
8: $G_i^{\#}$ = SCRA(G_i')
9: **end while**
10: **if** ($\exists$ $N_u \in G_i^{\#}$ && N_u.tag!=1) **then**
11: G_i^*=DCRA ($G_i^{\#}$)
12: **else**
13: G_i^*= $G_i^{\#}$
14: **end if**
15: $g^* \leftarrow G_i^*$
16: **end for**
17: return g^*

5 Experimental Results

This section analyzes and evaluates DSNDSA algorithm performance. The DSNDSA algorithm is compared with the dynamic k^w-number of mutual friend anonymity algorithm proposed by Jyothi [9] and the dynamic k^w-structure diversity anonymity algorithm proposed by Tai [17]. The experiment is tested using real social network datasets: Caida, Super User and wiki-talk. Among them, the Caida dataset is a relationships dataset that contains 122 CAIDA AS graphs from January 2004 to November 2007. The graph data of the network on 7 timestamps were obtained in the experiment. The Super User dataset is a temporal network of interactions on the stack exchange web site Super User. Edges (u, v, t) represents that user u answered user v's question at time t. The wiki-talk dataset is a temporal network representing Wikipedia users editing each other's Talk page.Edges (u, v, t) means that user u edited user v's talk page at time t. Dataset statistics is shown in Table 3.

In this paper, the directed graphs are processed to undirected graphs before the experiment. Experimental environment: CPU 1.80GHz, RAM 16GB, Hadoop 2.7.2,

Spark 2.4.3, programming language Scala 2.13.0 and 15 computing nodes.

Table 3: Dataset statistics

dataset	vertices	temporal edges	time span
Caida	26475	106762	122graphs
Super User	167981	430033	2773 days
wiki-talk	1140149	7833140	2320 days

5.1 Information Loss

In order to measure the influence of DSNDSA algorithm on graph structure in anonymous process, the average betweenness (BW) and average path length (APL) are used to evaluate the algorithm in the experiment. The BW is defined as the average of betweenness centrality of all vertices. The betweenness centrality of a vertexis calculated as Equation (1), where σ_{st} is the number of shortest paths from vertex s to vertex t and $\sigma_{st}(v)$ is the total number of those paths that pass through vertex v length of vertices u and v[16].

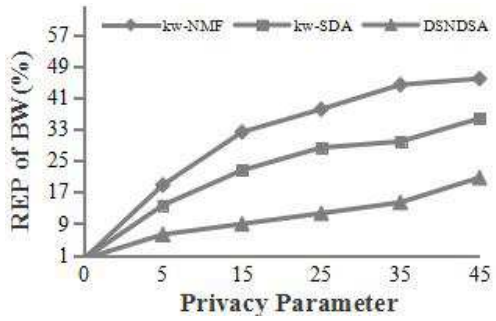
$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{1}$$

In order to express the information loss after anonymity intuitively, using the relative error percentage(REP) to measure the change of graph structure. As shown in Equation (2), G and G* represents the graph structure values in the original and anonymous social network graph respectively. The lower the value, the smaller the information loss.

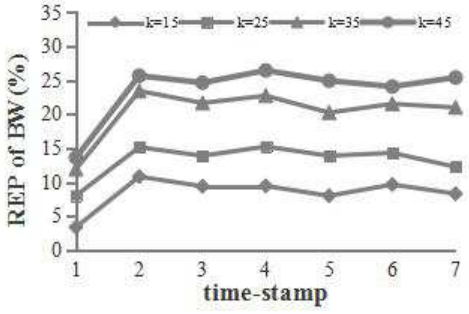
$$REP = \frac{|G - G^*|}{|G|} \times 100\% \tag{2}$$

Figure 7(a) shows the relative error percentage of BW after anonymity. With the increase of privacy parameter k, the relative error percentage curves of BW corresponding to different methods shows an upward trend, this means that the loss of information increases with the degree of privacy protection. The relative error percentages of k^w-NMF algorithm and k^w-SDA algorithm are both larger than DSNDSA algorithm after anonymity. Overall, DSNDSA algorithm has the best effect in ensuring the utility of graph structure.

Figure 7(b) shows the BW on each group data of Caida dataset, *i.e.* the relative error percentage of BW with time t under different k settings. Six groups of anonymous graphs are obtained when t=7. The relative error percentage increases with the increase of k on six groups of anonymous graphs. When k is fixed, t = 1 corresponds to the smallest relative error percentage for single graph anonymity. The relative error percentage of BW does not change much when t is other value, this is because the scale of each group of dynamic social network increases slightly with the passage of time.



(a) Super User



(b) Caida

Figure 7: Average betweenness

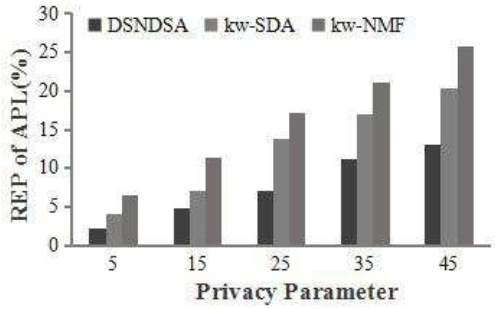
Figure 8(a) shows the average path lengths (APL) under different k settings on the Caida dataset. The low degree of privacy protection corresponds to the low relative error percentage of APL. The relative error percentage of APL corresponding to DSNSDSA algorithm is smaller, which shows that our method can better guarantee the structural properties of dynamic social network graph than k^w-NMF algorithm and k^w-SDA algorithm.

Figure 8(b) shows the relative error percentage of APL with time t on the Caida dataset. When k reaches 35, the relative error percentage of APL does not exceed 11.97.

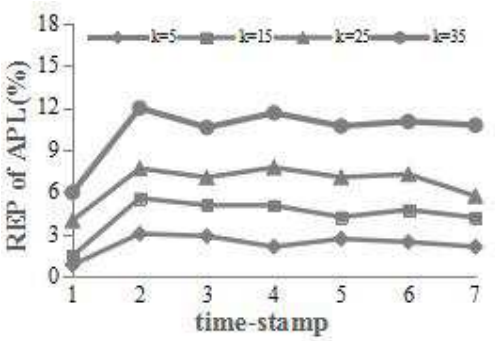
5.2 Availability of Community Structure

Suppose that the community sets in the original and anonymous social network are $C = \{C_1, C_2, \cdots, C_n\}$ and $C^* = \{C_1^*, C_2^*, \cdots, C_m^*\}$ respectively. Jaccard similarity coefficient can measure the similarity of social network community structure before and after anonymity. Jaccard similarity coefficient is expressed as a percentage. The greater its value, the higher the degree of protection of anonymity algorithm to the community structure of the original social network.

Figure 9(a) shows the variation of Jaccard similarity coefficient with k on wiki-talk dataset. And the vertical axis reflects the degree to which the anonymous graph preserves the original community structure. With the increase of k, the Jaccard similarity coefficients corresponding to the three methods shows a downward trend. For the case of larger k, DSNDSA algorithm protects the community structure more than 69.6%. Compared with the other two algorithms, it has better effect in community struc-

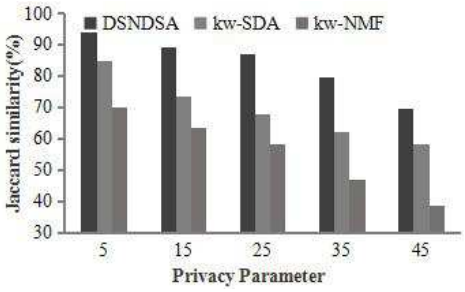


(a) Caida

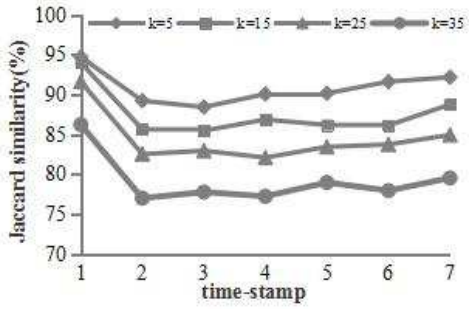


(b) Caida

Figure 8: Average path lengths



(a) wiki-talk



(b) Caida

Figure 9: Jaccard similarity coefficient

ture protection. Figure 9(b) shows the variation of Jaccard similarity coefficient with time t on Caida dataset. It can be observed that in the process of anonymity for two consecutive moments, the degree of protection of the anonymous graph to the original community structure will not decrease significantly with the increase of k.

Normalized Mutual Information(NMI) can measure the similarity between the community detection results of the algorithm and the real results. The paper takes the result of social network community detection before anonymity as the real result and compares it with the community structure after anonymity. Assuming that x and y represents two specific division results of the network respectively. The greater the NMI, the more information x and y can provide to each other and the closer they are. The calculation is shown in Equation (3).

$$U(X,Y) = \frac{2I(X,Y)}{H(X)+H(Y)} \tag{3}$$

Where,

$$\begin{aligned} I(X,Y) =& H(X) - H(X|Y) \\ =& \sum_{y\in Y}\sum_{x\in X} p(x,y)\log(\frac{p(x,y)}{p(x)p(y)}) \end{aligned} \tag{4}$$

p(x,y) represents the joint distribution probability of x and y, and adjusts the mutual information to 0-1 with an expected value of 1.

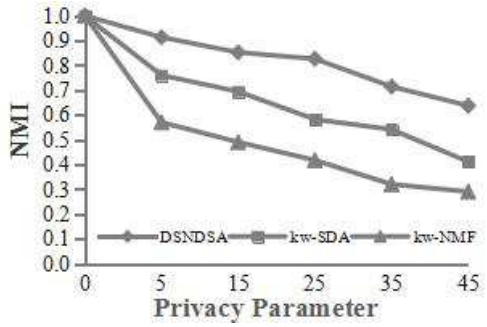
Figure 10(a) shows the variation of NMI with k on Super User dataset. With the increase of k, the NMI value tends to decrease, but the overall value is close to 1. This shows that DSNDSA algorithm has higher data availability than other algorithms in terms of community structure. Figure 10(b) shows the variation of NMI with time t on Caida dataset. We observe that each group of anonymous graphs can well protect the original social network community structure.

The precision index [2] can measure the change of the community to which the vertex belongs in the anonymity process. The precision index can be defined as Equation (5). If the community to which the vertex belongs remains unchanged after anonymity, the value of $\rho_{l_{tv}(v)=l_{pv}(v)}$ is 1; On the contrary, the value of $\rho_{l_{tv}(v)=l_{pv}(v)}$ is 0. The precision index is a value in range [0, 1]. The higher the value, the higher the usability of the community structure.

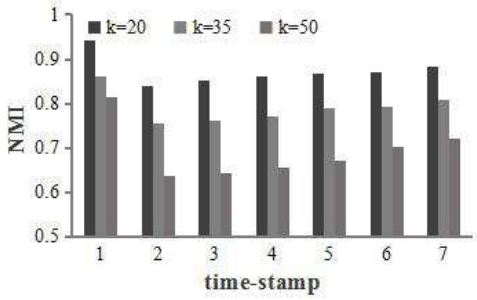
$$Precision\ index = \frac{1}{n}\sum\nolimits_{v\in G}\rho_{l_{tv}(v)=l_{pv}(v)} \tag{5}$$

Figure 11(a) shows the variation of precision index with k on Super User dataset. The precision index value decreases with the increase of k. Since DSNDSA algorithm considers the community structure in the process of anonymity, so the precision index is close to 1 after anonymity. The DSNDSA algorithm is better than the other two algorithms in terms of community structure availability.

Figure 11(b) shows the variation of precision index with time t on Caida dataset. When k is fixed and t is not less than 2, the precision index changes little, because the size of each group of dynamic networks increases slightly with time.

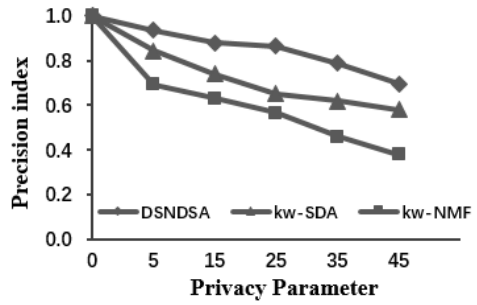


(a) Super User

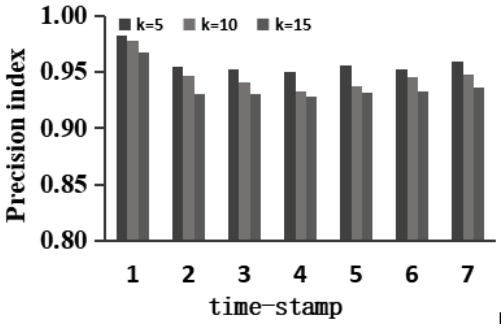


(b) Caida

Figure 10: Normalized mutual information



(a) Super User



(b) Caida

Figure 11: Precision index

6 Conclusion

The proliferation of online network data leads the dynamic network analysis and related privacy issues to become more important. This paper studies the privacy protection of dynamic social networks. The paper defines a vertex degree sequence attack model for dynamic social networks and proposes a distributed k-anonymity algorithm for dynamic social networks. The algorithm constructs a compressed binary tree to obtain vertex anonymity degree sequence, and adds dummy vertices to obtain dynamic social network anonymious graph. In addition, the algorithm merges dummy vertices in parallel based on the community to which the vertices belongs in order to improve the usability of the published graph in community structure. Experiments on real datasets shows that the algorithm in this paper can prevent the identity disclosure of dynamic social network vertices effectively while preserving the social network community structure and other graph structure properties, such as average betweenness, average path length, etc.

Acknowledgments

This work is partially supported by Natural Science Foundation of China (No.61562065) and Natural Science Foundation Project of Inner Mongolia (No.2019MS06001). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] A. Bhardwaj, V. Avasthi, and S. Goundar, "Impact of social networking on indian youth - a survey," *International Journal of Electronics and Information Engineering*, vol. 7, no. 1, pp. 41–51, 2017.

[2] B. J. Cai, H. Y. Wang, H. R. Zheng, and H. Wang, "Evaluation repeated random walks in community detection of social networks," *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 1849–1854, 2010.

[3] C. P. Cao and X. Zheng, "Research of anonymity model for privacy-preserving in social network," *Journal of Chinese Computer Systems*, vol. 37, no. 8, pp. 1821–1825, 2016.

[4] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, "k-degree anonymity and edge selection: Improving data utility in large networks," *Knowledge and Information Systems*, vol. 50, no. 2, pp. 447–474, 2017.

[5] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, "A survey of graph-modification techniques for privacy-preserving on networks," *Artificial Intelligence Review*, vol. 47, no. 3, pp. 341–366, 2017.

[6] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in com-

plex networks," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 4, no. 5, pp. 512–546, 2011.

[7] C. H. Guo, B. Wang, H. J. Zhu, and X. C. Yang, "Incremental dynamic social network anonymity technology," *Journal of Computer Research and Development*, vol. 53, no. 6, pp. 1352–1364, 2016.

[8] X. Y. Hu, L. Wang, J. Q. Tang, C. Lei, P. Liu, and X. X. Li, "Anonymizing approach to resist label-neighborhood attacks in dynamic releases of social networks," in *IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom'17)*, pp. 1–6, 2017.

[9] V. Jyothi and V. V. Kumari, "Privacy preserving in dynamic social networks," in *Proceedings of the International Conference on Informatics and Analytics*, pp. 1–8, 2016.

[10] M. Kiabod, M. N. Dehkordi, and B. Barekatain, "Tsram: A time-saving k-degree anonymization method in social network," *Expert Systems with Applications*, vol. 125, pp. 378–396, 2019.

[11] S. Kumar and P. Kumar, "Upper approximation based privacy preserving in online social networks," *Expert Systems with Applications*, vol. 88, pp. 276–289, 2017.

[12] X. Y. Liu, B. Wang, and X. C. Yang, "Survey on privacy preserving techniques for publishing social network data," *Journal of Software*, vol. 25, no. 3, pp. 576–590, 2014.

[13] Z. Y. Liu and Y. H. Ma, "A divide and agglomerate algorithm for community detection in social networks," *Information Sciences*, vol. 482, pp. 321–333, 2019.

[14] T. H. Ma, Y. Hao, X. F. Suo, Y. Xue, and J. Cao, "A weighted collaboration network generalization method for privacy protection in c-dblp," *Intelligent Data Analysis*, vol. 22, no. 1, pp. 3–19, 2018.

[15] K. R. Macwan and S. J. Patel, "k-degree anonymity model for social network data publishing," *Advances in Electrical and Computer Engineering*, vol. 17, no. 4, pp. 117–125, 2017.

[16] K. R. Macwan and S. J. Patel, "k-nmf anonymization in social network data publishing," *The Computer Journal*, vol. 61, no. 4, pp. 601–613, 2018.

[17] C. H. Tai, P. J. Tseng, S. Y. Philip, and M. S Chen, "Identity protection in sequential releases of dynamic networks," *IEEE transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 635–651, 2013.

[18] C. L. Wang, E. T. Wang, and A. L. Chen, "Anonymization for multiple released social network graphs," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 99–110, 2013.

[19] H. J. Wang, P. Liu, S. Lin, and X. X. Li, "A local-perturbation anonymizing approach to preserving community structure in released social networks," in *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pp. 36–45, 2016.

[20] S. L. Wang, Y. C. Tsai, T. P. Hong, and H. Y. Kao, "k(-)-anonymization of multiple shortest paths," *Soft Computing: A Fusion of Foundations, Methodologies and Applications*, vol. 21, pp. 4215–4226, 2017.

[21] Y. Z. Wang, L. Xie, B. H. Zheng, and K. C. Lee, "High utility k-anonymization for social network publishing," *Knowledge and Information Systems*, vol. 41, no. 3, pp. 697–725, 2014.

[22] D. R. Yu, H. X. Zhao, L. Wang, P. Liu, and X. X. Li, "A hierarchical k-anonymous technique of graphlet structural perception in social network publishing," in *International Conference on Mobile, Secure, and Programmable Networking*, pp. 224–239, 2018.

[23] Y. Zhao and Z. J. Li, "Privacy management in social network data publishing with community structure," in *The International Conference on Healthcare Science and Engineering*, pp. 141–151, 2018.

Biography

Na Li, is a graduate student in the School of Information Engineering, Inner Mongolia University of Science & Technology.Her research interests include Social Network privacy protection.

Xiao-lin Zhang, received the PhD degree from the Northeastern University of China, Shenyang, in 2006. She is a professor in the School of Information Engineering, Inner Mongolia University of Science & Technology. Her research interests include database theory and information security, Cloud Computing and Social Network privacy protection.

Yong-ping Wang, received the Master's degree from Wuhan University of Technology in 2010. She is a lecturer in the School of Information Engineering, Inner Mongolia University of Science & Technology. Her research interests include Data privacy protection.

Jian Li, is a graduate student in the School of Information Engineering, Inner Mongolia University of Science & Technology. His research interests include Community discovery and Social Network privacy protection.

Li-xin Liu, is a PhD candidate at Renmin University of China and the Member of China Computer Federation. Her main research interests include privacy protection and blockchain.

Decentralizing Multi-Authority Attribute-Based Access Control Scheme with Fully Hidden Policy

Leyou Zhang[1], Juan Ren[1,2], Li Kang[1], and Baocang Wang[3,4]
(Corresponding author: Juan Ren)
School of Mathematics and Statistics, Xidian University, Xi'an 710126, China[1]
Science and Technology on Communication Security Laboratory, Chengdu 610041, China[2]
School of Information Engineering, Xuchang University, Xuchang 461000, China[3]
State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China[4]
Email: juaner_r@126.com
(Received Jan. 4, 2020; Revised and Accepted Sept. 8, 2020; First Online May 30, 2021)

Abstract

Decentralized multi-authority attribute-based encryption (MA-ABE) is considered a potential method to protect users' privacy in the cloud. However, most of the existing works cannot provide a complete solution since some vulnerabilities can be found in users' collusion, global identity (GID) leakage-resilience, and access policy hiding. In this paper, we focus on overcoming these shortcomings. At first, we investigate the recent works and give a summary of them. Then an efficient decentralized MA-ABE scheme with a fully hidden access policy is presented. To implement the hidden access policy, we use the technique of Inner-Product Encryption (IPE). Under this technique, the Viète's Formulas is used to convert the access policy into a vector, which results in an efficient decentralized MA-ABE scheme with the shortened ciphertext and secret keys, which are only concerned with the number of wildcards. To further improve efficiency, the decryption is partially outsourced. The security of the proposed scheme is reduced to the standard decisional bilinear Diffie-Hellman (DBDH) assumption and the Decisional Linear (DLIN) assumption instead of other strong assumptions. Finally, performance analysis and numerical experiments confirm the scalability and flexibility of our approach.

Keywords: Decentralizing ABE; Fully Hidden Policy; IPE; Resistant-Collusion

1 Introduction

Cloud computing has been widely concerned, and continually developed at present because of its low cost, strong computing capacity, large storage capacity and high data security performance, which makes it convenient and profitable for data owners to share data on third-party cloud storage servers. Therefore, more individuals and enterprises upload application data to cloud storage servers. However, in many applications, the data owners hope that only authorized users can share their data. Additionally, the data owners cannot obtain the prior knowledge of who will share their data. Hence an access control policy is required for encrypted data in the cloud [6,16]. The attribute-based encryption (ABE) due to Sahai and Waters [24] provided a solution to the above problem, which it supported the fine-grained access control by encrypting data with various access policies.

1.1 Motivations

It can be found that attribute-based encryption is applied in cloud computing by summarizing the relevant work, but there are still many problems that need to be solved urgently as stated below:

Firstly, the most existing schemes are based on one authority. However, in real life, it is impractical and overburdened for one authority to authenticate and certificate all attributes. Therefore, single authority has been a bottleneck in a large system. With the development of cloud storage, there is more than one party to act as an authority. Hence, MA-ABE addresses this problem. However, in recent years, some MA-ABE schemes have been successfully attacked repeatedly as in [8, 26, 29] by means of collusion attacks, test attacks or logical attacks. How to further enhance the collusion-resistance of a decentralized MA-ABE scheme is still a subject worth studying.

Secondly, in a real cloud storage environment, the access policy itself could be sensitive information about users' attributes and be showed in the ciphertext, which will result in the leakage of users' sensitive information when the users want to upload the file encrypted by the access policy to the cloud storage server. Considering this example: An enterprise may release a number of specific files encrypted by the access policy: *(Leader* $\wedge$ *apartment A)* $\vee$ *(Secretary* $\wedge$ *apartmet B)*, notice that itself reveals

user's private attributes. It is significant to hide the access policy since it may lead to the privacy leakage. In the study of hiding access policy, most ABE schemes only realize partially hidden access policy. In this paper, another method to hide access policy is considered by combining IPE technology with ABE scheme, in which the user's attribute set is sent to the attribute authorities (AAs) in form of fuzzy vector based on IPE technology, so that AAs cannot know the specific information about attribute names or attribute values. But it is difficult to combine the IPE technology with ABE schemes.

Thirdly, in the existing MA-ABE schemes, exponentiation and pairing operations increase linearly with the number of attributes in the decryption phase, which leads to the increase of decryption costs. Improving the decryption efficiency is also a considerable challenge.

1.2 Our Contributions

As mentioned above, in recent years, many MA-ABE schemes have been attacked repeatedly and successfully by means of collusion attacks or test attacks. In addition, most of the existing MA-ABE schemes only realize partially hidden access policy, and there are compromises in efficiency simultaneously. In this paper, an efficient decentralized MA-ABE scheme with fully hiding access policy and collusion-resistant strongly is presented. Main contributions are summarized as follows:

Strong resistance to attacks. We propose a decentralized MA-ABE scheme with strong resistance to attacks from potential malicious users. Specifically, GID is coupled non-linearly with parameters f_1, f_2 and η_k in the secret keys to resist the attacks mentioned in [23] and [29].

Fully hidden policy. In order to achieve fully policy-hiding, we build the decentralized MA-ABE scheme using the technique of IPE. Based on Viète's Formulas, the access policy, consisting of the position of the symbols, is fully hidden by converting it into a vector.

Low overhead. The length of ciphertext and secret keys is shortened in our scheme, due to it is only related to the number of wildcards in the access policy. To further improve efficiency, the decryption is partially outsourced.

1.3 Paper Organization

We present the related works in Section 2. In Section 3, some preliminaries including the statements of bilinear map, complexity assumptions, access structure and the Viète's formulas are provided. Then the formal definition and its security model are given in Section 4. Section 5 presents the construction of our scheme in detail. The security analysis and performance analysis are proposed in Section 6 and Section 7 respectively. Finally, we give a brief conclusion in Section 8.

2 Related Works

We analyze related researches from three aspects: Multi-authority ABE, policy-hiding ABE and outsourcing ABE. The details are given as follows.

2.1 Multi-authority ABE

It started with the one by Chase [3] with a central authority (CA) and global identify (GID), which GID prevented the collusion attacks from malicious user. But it is limited to the AND-gate policy. Müller *et al.* [18] proposed the other one with CA and could be expressed by the LSSS access structure. However, the CA is required must be honest in [3, 18]. Then Chase and Chow introduced a new scheme that the center was removed [4]. However, The cooperation among multiple authorities is necessary during the system initialization phase. Later, Lewko and Waters [15] proposed a decentralized MA-ABE, in which the CA was removed so that any authorities could join or leave the system freely without reinitializing the system.

In addition, for the MA-ABE scheme, the most basic requirement is the resistance to collusion attacks. Hence, Han *et al.* [8] proposed a decentralized KP-ABE scheme that GID was non-linearly embedded into the user's private keys for enhancing the resistance to collusion attacks. Soon, Ge *et al.* [10] showed a new method of user's collusion attack, and proved the scheme [8] was vulnerable to this collusion attack. Compared with the previous proposed schemes, Han *et al.* [8] proposed a more powerful privacy protection MA-ABE scheme. However, Wang *et al.* [26] pointed out the security weaknesses above scheme and proposed a collusion attack method to Han's scheme. Qian *et al.* [22] constructed another multi-authority ciphertext-policy ABE (MA-CP-ABE) scheme that based on AND-gates access policy on multi-valued attributes. For this method of collusion attack mentioned in [22], Rahulamathavan *et al.* [23] proposed a decentralized ABE scheme that resisted it in 2016, such that the key generation algorithm was improved for breaking the linear relationship between keys. However, this scheme [23] was found that it could not resist the user collusion attacks, and the improved algorithm was given by Zhang *et al.* [29] in 2018. However, these schemes do not consider the feature of hiding policy.

2.2 Policy-Hiding ABE

Hiding policy (or attribute) means that the privacy in access policy is protected in the applications. Fully policy-hiding means that anyone could not know the sensitive attribute information from the access policy, even authorized users who could decrypt successfully.

Nishide *et al.* [19] introduced firstly the concept of policy-hiding by AND-gate access policy on multi-valued attributes with wildcards in 2008. However, the scheme is only proven in a weak model. Later, To protect sensitive information included in the access policy, several

Table 1: The comparison of our scheme and related works

Scheme	Multi-authority	Hidden policy	Way of policy-hiding	Outsource
[26]	Multi	✗	✗	✗
[29]	Multi	✗	✗	✗
[28]	Single	Partially Hidden	Hide attribute values	✗
[20]	Single	Fully Hidden	attribute values as: +,−,∗	✗
[31]	Single	Fully Hidden	Multi-valued attributes	✗
[21]	Single	Fully Hidden	IPE	✗
[7]	Multi	Partially Hidden	Multi-valued attributes	✗
[30]	Multi	Fully Hidden	One-way anonymous key agreement	✗
[1]	Multi	Fully Hidden	One-way anonymous key agreement	✓
[17]	Multi	Fully Hidden	Randomizing-polynomial encodings	✗
[25]	Multi	✗	✗	✓
Ours	Multi	Fully Hidden	IPE + position of attribute(+,−,∗)	✓

[1] + or − respectively refers to whether an attribute exists on the access policy or not.
[2] ∗ means that an attribute can be either positive or negative attributes.

ABE schemes with partially hidden access policy were proposed [5, 14, 28]. In most of them, each attribute in the access policy is represented as a couple: The attribute name and the attribute value. Generically, the attribute values contain more sensitive information. For example, the attribute values "secretary" and "CN2019" are more sensitive than the attribute names "Position" and "ID Number", respectively. The above ABE schemes protect the sensitive information by hiding the attribute values. However, the attribute names are revealed in the access policy $(Position : \star) \wedge (ID\ Number : \star)$. Therefore, there are a set of security issue in [5, 14, 28], especially the off-line dictionary attacks on partially hidden access policy.

To address the security issues raised by ABE schemes with partially hidden access policy, ABE schemes with fully hidden access policy were introduced in [20, 21, 27, 31]. Xu *et al.* [20] extended the ABE scheme due to Bethencourt *et al.* [2], and proposed an ABE scheme with hidden access policy based on the tree-like access policy for cloud applications, in which the value of each attribute could be represented by three kinds of symbols: $+$, $-$, $*$. However, this scheme relies on only one authority to manage the private keys, so the center authority must be honest and overburdened. In 2015, Zhou *et al.* [31] introduced a privacy preserving attribute-based broadcast encryption scheme with an expressive hidden access policy. However, this construction introduces a high computation because of much pairing operations. In 2016, Phuong *et al.* [21] proposed a new hidden access policy ABE scheme under standard assumptions. Their scheme is based on the IPE and realizes the policy hiding by representing the attributes in the access policy with the position of symbols. Later, Jin *et al.* [12] extends Phuong's scheme to be fully secure one.

Most of the mentioned schemes either fail to consider the feature of hiding policy or are single-authority ABE. Recently, to solve these problems, some MA-ABE schemes with hidden access policy were presented [1, 7, 17, 30]. In 2016, Zhong *et al.* [30] proposed the first policy hidden ABE scheme using multiple attribute authorities architecture. However, the exponential computing cost is required during the decryption due to pairing operations. In 2017, Fan *et al.* [7] presented a MA-CP-ABE access control scheme with hidden policy and constant length ciphertext. But this scheme relies on a weaker model which is called weakly policy (attribute)-hiding. Under this model, a party might decrypt the received ciphertext but the policy is remained unknown to any users, which means the policy may be leaked only upon the final successful decryption. In 2018, Belguith *et al.* [1] proposed a securely outsourcing MA-ABE scheme based on LSSS with hidden policy for cloud assisted IoT. However, it is proven be selectively secure. Recently, Michalevsky et al proposed a full policy-hiding ABE based on IPE [17]. It supports conjunctions, disjunctions and threshold policies and protects the access policies from any user and party that are not authorized to recover the messages. However, this scheme needs coordinations among the authorities at the beginning of Setup algorithm. Additionally, their scheme relies on the random oracle and is reduced to the SXDH assumption and k-Lin assumption.

2.3 Outsourcing ABE

In most of the existing policy-hiding ABE schemes, The decryption computation costs grow proportionally with complexity of the access policy. Hence, many works solves them by using the outsourcing decryption method [9, 11, 13]. In 2017, Shao *et al.* introduced this method to decentralized MA-ABE to decrease the decryption cost [25]. However, their scheme relies on the random oracle and do not consider the hiding policy.

In conclusion, it is very urgent to propose a MA-ABE scheme with strong resistance to attacks, which can achieve the optimal compromise between privacy and efficiency. To evaluate the motivations given in introduction, we introduce a comparison of our scheme with other ABE constructions of recent years in Table 1, that are most closely-related to our scheme.

3 Techniques Preliminaries

To make the description concise, we first give some symbols used in this paper and their meanings. The details

are shown in Table 2.

Table 2: Symbols used in this scheme

Symbol	Implication
U	The attribute universe in this system
S	The user's attribute list
W	The access policy
AA_k	The k-th attribute authority
AA_k^*	The k-th corrupted attribute authority
U_k	The attribute list managed by AA_k
S^k	$U_k \cap S$
N_1, N_2, N_3	The number of the symbols $+,-,*$ respectively
$\overrightarrow{x_{V^k}}, \overrightarrow{x_{Z^k}}$	Two vectors converted by S^k
$\overrightarrow{v}$	A vector converted by W
n	The length of the vector $\overrightarrow{v}$, $\overrightarrow{x_{V^k}}$ or $\overrightarrow{x_{Z^k}}$
pp	The system public parameters
PK_k/SK_k	The public key/ secret key of AA_k
GID	The user's global identity
$SK_{GID,k,i}$	The attribute secret key of the user GID from AA_k

3.1 Bilinear Map and Complexity Assumptions

Definition 1. *(Bilinear map): Let $\mathbb{G}$ and $\mathbb{G}_\mathbb{T}$ be two multiplicative cycle groups of same prime order p, g is the generator of $\mathbb{G}$. $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathbb{T}$ is a bilinear map with the following properties:*

1) *Bilinearity: $\forall a, b \in Z_p$ and $e(g^a, g^b) = e(g, g)^{ab}$.*
2) *Non-Degeneracy: $e(g, g) \neq 1$;*
3) *Computability: $e(g, g)$ is polynomial-time computable.*

Assumption 1. *(DBDH Assumption): Let $a, b, c, z \in_R Z_p$. Given the tuple $(A, B, C)=(g^a, g^b, g^c)$, the DBDH assumption holds when no polynomial-time algorithm $\mathcal{B}$ can distinguish $e(g, g)^{abc}$ and $e(g, g)^z$ with non-negligible advantage. The advantage of algorithm $\mathcal{B}$ is*

$$Adv_{\mathcal{B}}^{DBDH} = |\Pr[\mathcal{B}(A, B, C, e(g, g)^{abc}) = 1] - \Pr[\mathcal{B}(A, B, C, e(g, g)^z) = 1]| \leq \epsilon. \quad (1)$$

Assumption 2. *(DLIN Assumption): Let z_1, z_2, z_3, z_4, $z \in_R Z_p$. Given the tuple $(Z_1, Z_2, Z_3, Z_4) = (g^{z_1}, g^{z_2}, g^{z_3+z_4}, g^{z_2 z_4})$, the DLIN assumption holds when no polynomial-time algorithm $\mathcal{B}$ can distinguish $g^{z_1 z_3}$ and g^z with non-negligible advantage. The advantage of algorithm $\mathcal{B}$ is*

$$Adv_{\mathcal{B}}^{DLIN} = |\Pr[\mathcal{B}(Z_1, Z_2, Z_3, Z_4, g^{z_1 z_3}) = 1] - \Pr[\mathcal{B}(Z_1, Z_2, Z_3, Z_4, g^z) = 1]| \leq \epsilon. \quad (2)$$

3.2 Access Policy

Consider the access policy based on AND-gates with wildcards. Let the attribute universe descriptions be $U = \{Att_1, Att_2, ..., Att_L\}$. The user's attribute list is denoted as $S = \{S_1, S_2, ..., S_L\}$ where each attribute S_i could be: $+$ or $-$. Let $W = \{S_1^\star, S_2^\star, ..., S_L^\star\}$ be an AND-gate access policy with wildcards where each attribute $S_i^\star$ could be: $+$, $-$ or $*$. The notation $S \models W$ means that the user's attribute list satisfies the access policy.

3.3 The Viète's Formulas

Consider two vectors $\vec{p} = (p_i)$ and $\vec{q} = (q_i), i = 1, ..., L$, where p_i could be alphabets or wildcards, and q_i is alphabets. $H = \{h_1, ..., h_n\} \subset \{1, ..., L\}$ is defined by the positions of the wildcards in vector $\vec{p}$.

Let $\prod_{h \in H}(i - h) = \sum_{k=0}^{n} \lambda_k i^k$, where λ_k are the coefficients dependent on H. If $p_i = q_i \vee p_i = *$:

$$\sum_{i=1, i \notin H}^{L} p_i \prod_{h \in H}(i - h) = \sum_{k=0}^{n} \lambda_k \sum_{i=1}^{L} q_i i^k \quad (3)$$

The coefficient λ_k can be constructed by the Viète's Formulas as follows, where $n = |H|$.

$$\lambda_{n-k} = (-1)^k \sum_{1 \leq i_1 < i_2 < ... < i_k \leq n} h_{i_1} h_{i_2} ... h_{i_k}, 0 \leq k \leq n$$

4 Formal Definition and Security Model

4.1 System Model

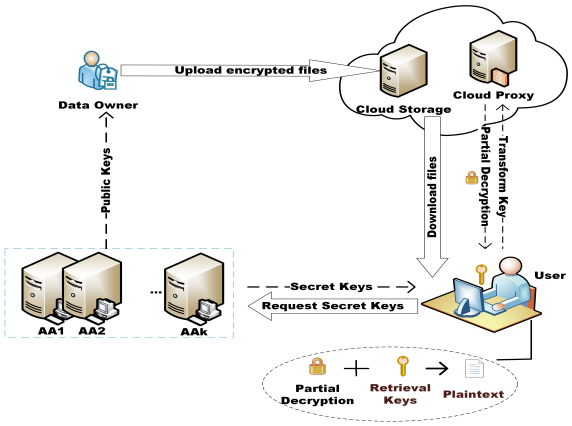


Figure 1: System model

There are five entities: Data owners (DO), data users (DU), several attribute authorities (AAs), and cloud server including cloud storage server(CS) and cloud proxy server(CP) in the system as showed in Figure 1, the details are as follows:

Step 1: In the system initialization stage, the public parameters are generated, and each AA generates the public keys and sends to DO.

Step 2: DO specifies the access policy and encrypts files using the public keys and the access policy, then uploads the encrypted files to CS, in which CS is used to store encrypted files and provide access services for DU.

Step 3: After DU downloads the encrypted file from CS, if DU wants to decrypt it, DU needs to request the secret keys to AAs. Notice that the encrypted file can be decrypted successfully by DU, if and only if DU's attribute list satisfies the access policy.

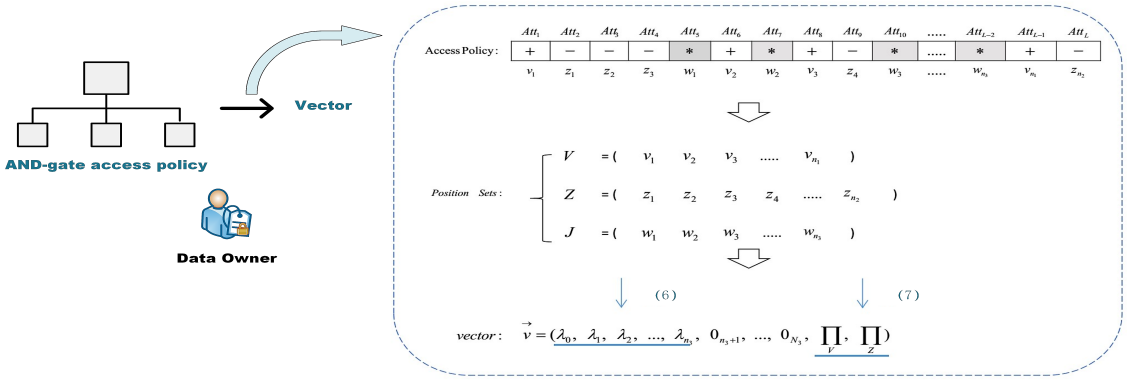


Figure 2: Convert the access policy into an vector

Step 4: After receiving the request of DU, each AA verifies DU's identity, then distributes the secret keys for legitimate DU.

Step 5: To reduce the burden of calculation, DU converts the secret keys into the transform keys to CP, and remains the retrieval keys. Then CP is responsible for partial decryption.

Step 6: Finally, DU can recover the plaintext using retrieval keys and the information of the partial decryption.

4.2 Scheme Definition

The scheme consists of seven algorithms as follows:

Global setup ($1^\lambda \rightarrow pp$): The system is produced at this stage. It inputs security parameters λ, and returns the public parameters pp.

Authority setup (pp,$k \rightarrow PK_k$,SK_k): It inputs pp, and the authority index k, then it outputs the authority's public keys PK_k and secret keys SK_k.

Encryption (pp,W,M,$PK_k \rightarrow CT$): It inputs pp, the public key PK_k, the message M, and the access policy W, then outputs the ciphertext CT to CS.

KeyGen (pp,SK_k,GID,$S \rightarrow SK_{GID,k,i}$): It takes SK_k, GID, attributes set S as input, returns the secret keys $SK_{GID,k,i}$ to DU.

TransKeyGen($pp, SK_{GID,k,i} \rightarrow TK_{GID,k,i}, RK_{GID,k,i}$)**:** It takes pp and $SK_{GID,k,i}$ as input, then returns transformation keys $TK_{GID,k,i}$ to CP and retains a retrieval key $RK_{GID,k,i}$ to DO.

Out.Decryption (pp,CT,$TK_{GID,k,i} \rightarrow \widehat{CT}$): It inputs pp, CT, and $TK_{GID,k,i}$, then returns $\widehat{CT}$ to DU.

User.Decryption (pp,$\widehat{CT}$,$RK_{GID,k,i} \rightarrow M$): It takes $\widehat{CT}$ and $RK_{GID,k,i}$ as input, then outputs the recovered M.

4.3 Security Model

Based on DBDH and DLIN assumption, the scheme is proven to be the selective IND-CPA security by the security game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$. The details are as following:

Initialization: $\mathcal{A}$ submits two challenge access structures W_0, W_1 and a series of corrupted authorities AA_k^* to $\mathcal{C}$, where $|AA_k^*| < \mathcal{K}$.

Global Setup: $\mathcal{C}$ runs the *Global Setup* algorithm and outputs pp to $\mathcal{A}$.

Authorities Setup:

1) For the corrupted authorities, $\mathcal{C}$ sends PK_k and SK_k to $\mathcal{A}$.
2) For the honest authorities, $\mathcal{C}$ sends PK_k to $\mathcal{A}$.
3) For the half-honest authorities, $\mathcal{C}$ sends PK_k and parts of SK_k to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ submits the attribute set S and GID for querying secret keys. If ($S \models W_0 \wedge S \models W_1$) or ($S \not\models W_0 \wedge S \not\models W_1$), $\mathcal{C}$ sends SK_S to the adversary. $\mathcal{A}$ can query polynomially.

Challenge: $\mathcal{A}$ submits two equal length messages M_0 and M_1. $\mathcal{C}$ flips a random coin ξ and runs the Encryption algorithm. $\mathcal{C}$ sends CT_ξ to $\mathcal{A}$. Note that if $\mathcal{A}$ obtains SK_S under the condition ($S \models W_0 \wedge S \models W_1$) in Phase 1, then it is needed that $M_0 = M_1$.

Phase 2: Phase 1 is repeated. If $M_0 \neq M_1$, $\mathcal{A}$ can't submit S' such that $S' \models W_0 \wedge S' \models W_1$.

Guess: Finally, $\mathcal{A}$ outputs his guess ξ' on ξ.

Definition 2. *The decentralized ABE scheme with fully hidden policy is selective IND-CPA security if against any probabilistic polynomial-time adversary $\mathcal{A}$,*

$$Adv_{\mathcal{A}}^{IND-CPA}(\lambda) = |\Pr[\xi' = \xi] - \frac{1}{2}| \quad (4)$$

is negligible in the security parameter λ.

5 Our Construction

5.1 Extending Phuong's Technology

In our scheme, we also extend Phoung's technique [21] to convert an access policy into a vector $\overrightarrow{v}$ which is combined with the technique of IPE to encrypt the data. In addition, for each authority, the attribute set S^k is converted into two vectors $\overrightarrow{x_{V^k}}, \overrightarrow{x_{Z^k}}$ which is used in key generation. Again, the conversion is performed by combining with the Viète's formulas and the positions of symbols. The details are showed as following:

5.1.1 Convert the Access Policy into an Vector

Firstly, the access policy W that consists of $+$, $-$, and $*$ can be separated into three position sets: V, Z, and J, which contains the positions of $+$, $-$, and $*$ in W respectively, where let $V = \{v_1, ..., v_{n_1}\}, Z = \{z_1, ..., z_{n_2}\}, J = \{w_1, ..., w_{n_3}\}$ ($n_i \leq N_i$, i=1, 2, 3). Next, based on the position set J and the Viète's formulas, we can calculate the coefficients $(\lambda_0, \lambda_1, ..., \lambda_{n_3})$, as $\lambda_{n_3} = 1, \lambda_{n_3-1} = -(w_1 + w_2... + w_{n_3}), \lambda_{n_3-2} = (w_1w_2 + w_1w_3 + ... + w_{n_3-1}w_{n_3}), ..., \lambda_0 = -(w_1w_2...w_{n_3})$.

And construct a polynomial $\sum_{k=0}^{n_3} \lambda_k i^k$, where i is the position of $+$ or $-$. Then we combine V and Z respectively as follows:

$$\begin{aligned} \prod{}_V &= +\sum\nolimits_{v_i \in V} \prod\nolimits_{w_j \in J} (v_i - w_j), \\ \prod{}_Z &= -\sum\nolimits_{z_i \in Z} \prod\nolimits_{w_j \in J} (z_i - w_j). \end{aligned} \tag{5}$$

Finally, we can convert the access policy W into a vector

$$\begin{aligned} \overrightarrow{v} &= (v_1, v_2, ..., v_n), \\ &= (\lambda_0, \lambda_1, ..., \lambda_{n_3}, 0_{n_3+1}, ..., 0_{N_3}, \prod{}_V, \prod{}_Z). \end{aligned} \tag{6}$$

where N_1, N_2, $N_3 \leq L$ show the maximum number of $+$, $-$, and $*$ in an access policy respectively. The process is shown in Figure 2.

5.1.2 Convert the Attributes set S^k into Two Vectors

In user key generation, attributes set S^k containing $+$ and $-$ attributes also need to be separated into two sets V^k and Z^k which contains respectively positions of positive and negative attributes. Then calculate:

$$v_l^* = -\sum_{v_i^k \in V^k} v_i^{kl},\ z_l^* = +\sum_{z_i^k \in Z^k} z_i^{kl}\ (l = 0, ..., N_3). \tag{7}$$

Finally, the attributes set S^k is converted into two vectors:

$$\begin{aligned} \overrightarrow{x_{V^k}} &= (x_{V_1^k}, x_{V_2^k}, ..., x_{V_n^k}) = (v_0^*, v_1^*, ..., v_{N_3}^*, 1/\mathcal{K}, 0), \\ \overrightarrow{x_{Z^k}} &= (x_{Z_1^k}, x_{Z_2^k}, ..., x_{Z_n^k}) = (z_0^*, z_1^*, ..., z_{N_3}^*, 0, 1/\mathcal{K}). \end{aligned} \tag{8}$$

The process is shown in Figure 3, in which we assume (Att_1, Att_2, Att_3)$\subseteq U_1$; (Att_4, Att_5, Att_6, Att_7)$\subseteq U_2$; (Att_8, Att_9, Att_{10})$\subseteq U_3$; and so on; (Att_{L-2}, Att_{L-1}, Att_L)$\subseteq U_{\mathcal{K}}$;where U_k is the attribute set be managed by authority A_k. In summary, $(\overrightarrow{v}, \sum_{k=1}^{\mathcal{K}} \overrightarrow{x_{V^k}}) = 0$, $(\overrightarrow{v}, \sum_{k=1}^{\mathcal{K}} \overrightarrow{x_{Z^k}}) = 0$ iff $v_i = v_i^* \vee v_i = *$ and $z_i = z_i^* \vee z_i = *$, since combining Figure 2 and Figure 3, calculating:

$$\begin{aligned} &(\overrightarrow{v}, \sum\nolimits_{k=1}^{\mathcal{K}} \overrightarrow{x_{V^k}}) \\ &= -(v_1^{*0} + v_2^{*0} \cdots + v_{n_1}^{*\ 0}) \cdot \lambda_0 - (v_1^{*1} + v_2^{*1} \cdots \\ &\quad + v_{n_1}^{*\ 1}) \cdot \lambda_1 \cdots - (v_1^{*N_3} + v_2^{*N_3} \cdots + v_{n_1}^{*\ N_3}) \\ &\quad \cdot \lambda_{N_3} + \prod{}_V \cdot (\sum\nolimits_{k=1}^{\mathcal{K}} 1/\mathcal{K}) \\ &= -(v_1^{*0} + v_2^{*0} \cdots + v_{n_1}^{*\ 0}) \cdot \lambda_0 - (v_1^{*1} + v_2^{*1} \cdots \\ &\quad + v_{n_1}^{*\ 1}) \cdot \lambda_1 \cdots - (v_1^{*N_3} + v_2^{*N_3} \cdots + v_{n_1}^{*\ N_3}) \\ &\quad \cdot \lambda_{N_3} + (v_1^0 + v_2^0 \cdots + v_{n_1}^0) \cdot \lambda_0 + (v_1^1 + v_2^1 \cdots \\ &\quad + v_{n_1}^1) \cdot \lambda_1 \cdots + (v_1^{N_3} + v_2^{N_3} \cdots + v_{n_1}^{N_3}) \cdot \lambda_{N_3}. \end{aligned} \tag{9}$$

$$\begin{aligned} &(\overrightarrow{v}, \sum\nolimits_{k=1}^{\mathcal{K}} \overrightarrow{x_{Z^k}}) \\ &= +(z_1^{*0} + z_2^{*0} \cdots + z_{n_2}^{*\ 0}) \cdot \lambda_0 + (z_1^{*1} + z_2^{*1} \cdots \\ &\quad + z_{n_2}^{*\ 1}) \cdot \lambda_1 \cdots + (z_1^{*N_3} + z_2^{*N_3} \cdots + z_{n_2}^{*\ N_3}) \\ &\quad \cdot \lambda_{N_3} - \prod{}_Z \cdot (\sum\nolimits_{k=1}^{\mathcal{K}} 1/\mathcal{K}) \\ &= +(z_1^{*0} + z_2^{*0} \cdots + z_{n_2}^{*\ 0}) \cdot \lambda_0 + (z_1^{*1} + z_2^{*1} \cdots \\ &\quad + z_{n_2}^{*\ 1}) \cdot \lambda_1 \cdots + (z_1^{*N_3} + z_2^{*N_3} \cdots - z_{n_2}^{*\ N_3}) \\ &\quad \cdot \lambda_{N_3} - (z_1^0 + z_2^0 \cdots + z_{n_2}^0) \cdot \lambda_0 - (z_1^1 + z_2^1 \cdots \\ &\quad + z_{n_2}^1) \cdot \lambda_1 \cdots - (z_1^{N_3} + z_2^{N_3} \cdots + z_{n_2}^{N_3}) \cdot \lambda_{N_3} \end{aligned}$$

5.2 Decentralizing Attribute-Based Access Control Scheme With Fully Hidden Policy

The algorithm of our scheme is presented as follows:

Global Setup: Given the security parameter λ, the algorithm returns a bilinear group $param = (p, g, e, \mathbb{G}, \mathbb{G}_{\mathbb{T}})$. Let $H : \{0,1\}^* \to \mathbb{G}$ be a hash function, and $n = N_3 + 3$. Defining that there are $\mathcal{K}$ authorities in the system, and each authority AA_k manages disjoint attribute set $U_k = \{Att_1, Att_2, ..., Att_{n_k}\}$, where$|U_k| = n_k$. Later, it selects randomly $\{\Delta, f_1, f_2, \mu_1, \mu_2, \theta_1, \theta_2\} \in Z_p$, $g_2 \in \mathbb{G}$, then publishes the public parameters pp as follows:

$$\begin{aligned} pp = \{&param, V_1 = g^{\mu_1}, V_2 = g^{\mu_2}, \\ &X_1 = g^{\theta_1}, X_2 = g^{\theta_2}, g_1 = g^{\Delta}\} \end{aligned} \tag{10}$$

Authority Setup: The algorithm is run by AA_k as Algorithm 1.

Encryption: The algorithm is run by DO, the detailed process is as Algorithm 2.

KeyGen: DU submits $u = H(GID)$ and S to AA_k for requesting the secret keys. Each AA_k runs Algorithm 3 and distributes the secret keys to DU.

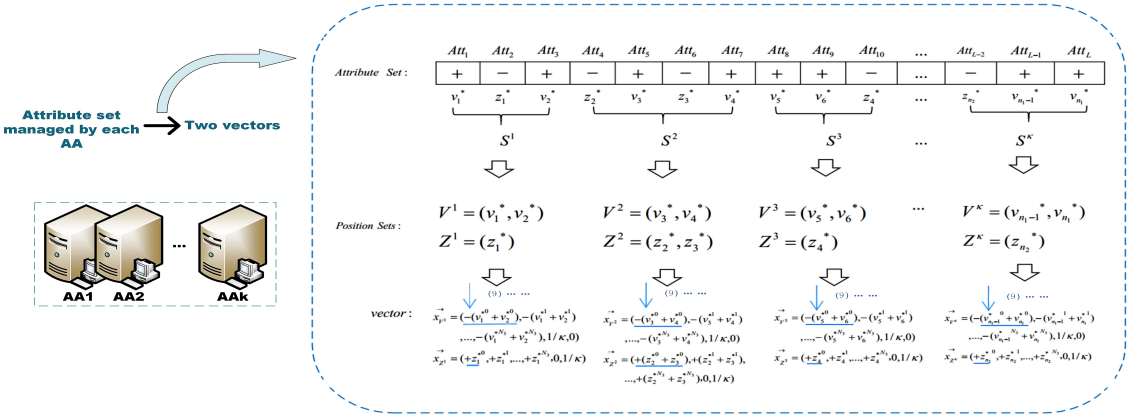


Figure 3: Convert the attribute set into two vectors

Algorithm 1 Authority Setup

Require: pp, k
Ensure: SK_k, PK_k
for each authority AA_k in system **do**
select α_k, γ_k, β_k, ζ_k, ς_k, η_k;compute: $T_k = g^{\gamma_k}$, $Z_k = g^{\beta_k}$, $M_k = g^{\zeta_k}$, $N_k = g^{\varsigma_k}$, $Y_k = e(g, g_2)^{\alpha_k}$;
for i in [1,n] **do**
select successively $u_{1,i,k}$,$w_{1,i,k}$,$u_{2,i,k}$,$w_{2,i,k}$ under following condition:
$\Delta = \mu_1 u_{2,i,k} - \mu_2 u_{1,i,k} = \theta_1 w_{2,i,k} - \theta_2 w_{1,i,k}$;
compute: $U_{1,i,k} = g^{u_{1,i,k}}$, $U_{2,i,k} = g^{u_{2,i,k}}$, $W_{1,i,k} = g^{w_{1,i,k}}$, $W_{2,i,k} = g^{w_{2,i,k}}$;
end for
end for
return PK_k=($\{Y_k, T_k, Z_k, M_k, N_k\}$, $\{U_{1,i,k}, U_{2,i,k}, W_{1,i,k}, W_{2,i,k}\}_{i=1}^{n}$) and SK_k=($\{\alpha_k, \gamma_k, \beta_k, \zeta_k, \varsigma_k, \eta_k\}$, $\{u_{1,i,k}, u_{2,i,k}, w_{1,i,k}, w_{2,i,k}\}_{i=1}^{n}$)$_{k=1}^{\mathcal{K}}$;

Algorithm 2 Encryption

Require: pp, PK_k, W, $M \in \mathbb{G}_{\mathbb{T}}$
Ensure: the ciphertext CT
for each data owner in system **do**
convert W into the vector $\overrightarrow{v}$ as subsection 5.1.1;
select $s_1, s_2, \beta \in Z_p$, compute $C_A = g^{s_2}$, $C_B = g_1^{s_1}$;
for k in [1,$\mathcal{K}$] **do**
compute: $C_0 = \prod_{k=1}^{\mathcal{K}} M \cdot e(g, g_2)^{\alpha_k s_2}$;
for i in [1,n] **do**
compute: $C_{1,i,k} = {U_{1,i,k}}^{s_1} \cdot {V_1}^{v_i \beta} \cdot T_k^{s_2}$,
$C_{2,i,k} = {U_{2,i,k}}^{s_1} \cdot {V_2}^{v_i \beta} \cdot Z_k^{s_2}$,
$C_{3,i,k} = {W_{1,i,k}}^{s_1} \cdot {X_1}^{v_i \beta} \cdot M_k^{s_2}$,
$C_{4,i,k} = {W_{2,i,k}}^{s_1} \cdot {X_2}^{v_i \beta} \cdot N_k^{s_2}$;
end for
end for
end for
return the ciphertext CT=(C_0,C_A,C_B,$\{C_{1,i,k}, C_{2,i,k}, C_{3,i,k}, C_{4,i,k}\}_{i=1}^{n}{}_{k=1}^{\mathcal{K}}$);

TransKeyGen: DU chooses a random number $z \in Z_p$, and constructs the transformation keys $TK_{GID,k,i}$ and the retrieval keys $RK_{GID,k,i}$ as follows. Note that $TK_{GID,k,i}$ is sent to CP, and $RK_{GID,k,i} = z$ is remained.

$$\begin{aligned} TK_{GID,k,i} =& (K_{A_k}{}^{\frac{1}{z}}, K_{B_k}{}^{\frac{1}{z}}, K_{1,i,k}{}^{\frac{1}{z}}, \\ & K_{2,i,k}{}^{\frac{1}{z}}, K_{3,i,k}{}^{\frac{1}{z}}, K_{4,i,k}{}^{\frac{1}{z}})_{i=1}^{n}{}_{k=1}^{\mathcal{K}} \\ =& (K_{A_k}{}', K_{B_k}{}', K_{1,i,k}{}', \\ & K_{2,i,k}{}', K_{3,i,k}{}', K_{4,i,k}{}')_{i=1}^{n}{}_{k=1}^{\mathcal{K}}. \end{aligned} \tag{11}$$

Out.Decryption: CP runs the Out.Decryption algorithm and calculates as follow:

$$CT_1 = \prod_{k=1}^{\mathcal{K}} \prod_{j=1}^{4} \prod_{i=1}^{n} e(C_{j,i,k}, K_{j,i,k}{}') \tag{12}$$

$$CT_2 = \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} [e(C_A, K_{A_k}{}') \cdot e(C_B, K_{B_k}{}')].$$

then returns $\widehat{CT} = \{CT_1, CT_2\}$ to DU.

User.Decryption: After obtaining $\widehat{CT}$ from CP, DU runs the *User.Decryption* algorithm and calculates as follows: $C_0/(CT_1 \cdot CT_2)^z = M$.

5.3 Correction Analysis

Calculate firstly as follow:

$$\begin{aligned} & e(C_{1,i,k}, K_{1,i,k}) \\ =& e({U_{1,i,k}}^{s_1} {V_1}^{v_i \beta} T_k^{s_2}, {V_2}^{-r_{1,i,k}} {U_{2,i,k}}^{\frac{x_{V_i^k}}{u+f_1}+\eta_k}) \\ =& e(g, K_{1,i,k})^{s_2 \gamma_k} \cdot e(g, g)^{s_1(-u_{1,i,k}\mu_2) r_{1,i,k}} \\ & \cdot e(g, g)^{(\frac{x_{V_i^k}}{u+f_1}+\eta_k) \cdot s_1 u_{1,i,k} u_{2,i,k}} \cdot e(g, g)^{-\mu_1 \mu_2 \beta v_i r_{1,i,k}} \\ & \cdot e(g, g)^{\beta(u_{2,i,k}\mu_1) v_i \cdot (\frac{x_{V_i^k}}{u+f_1}+\eta_k)} \end{aligned} \tag{13}$$

Algorithm 3 Key Generation

Require: SK_k, S, GID

Ensure: user's secret key $SK_{GID,k,i}$

for each authority AA_k in system **do**

convert S^k into vectors: $\overrightarrow{x_{V^k}}$,$\overrightarrow{x_{Z^k}}$ as subsection 5.1.2

for i in [1,n] **do**

select randomly $r_{i,k,1}, r_{i,k,2}$;

compute $K_{1,i,k} = V_2{}^{-r_{1,i,k}} \cdot U_{2,i,k}{}^{\frac{x_{V_i^k}}{u+f_1}+\eta_k}$,

$K_{2,i,k} = V_1{}^{r_{1,i,k}} \cdot U_{1,i,k}{}^{-(\frac{x_{V_i^k}}{u+f_1}+\eta_k)}$,

$K_{3,i,k} = X_2{}^{-r_{2,i,k}} \cdot W_{2,i,k}{}^{\frac{x_{Z_i^k}}{u+f_2}-\eta_k}$,

$K_{4,i,k} = X_1{}^{r_{2,i,k}} \cdot W_{1,i,k}{}^{-(\frac{x_{Z_i^k}}{u+f_2}-\eta_k)}$,

$K_{A_k} = g_2{}^{\alpha_k} \prod_{i=1}^{n} (K_{1,i,k}{}^{-\gamma_k} K_{2,i,k}{}^{-\beta_k} K_{3,i,k}{}^{-\zeta_k}$

$K_{4,i,k}{}^{-\varsigma_k})$, $K_{B_k} = \prod_{i=1}^{n} g^{-(r_{1,i,k}+r_{2,i,k})\triangle}$;

end for

end for

return the user's secret key $SK_{GID,k,i}$=(K_{A_k}, K_{B_k}, $\{K_{1,i,k}, K_{2,i,k}, K_{3,i,k}, K_{4,i,k}\}_{i=1}^{n}{}_{k=1}^{\mathcal{K}}$);

Then we have:

$$
\begin{aligned}
CT_1 &= \prod_{k=1}^{\mathcal{K}} \prod_{j=1}^{4} \prod_{i=1}^{n} e(C_{j,i,k}, K_{j,i,k})^{\frac{1}{z}} \\
&= [\prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{1,i,k})^{\gamma_k s_2} \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{2,i,k})^{\beta_k s_2} \\
&\cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{3,i,k})^{\zeta_k s_2} \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{4,i,k})^{\varsigma_k s_2} \\
&\cdot e(g,g)^{\sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} (r_{1,i,k}+r_{2,i,k}) s_1 \Delta} \\
&\cdot e(g,g)^{\sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} (\frac{x_{V_i^k} v_i \beta \Delta}{f_1+u} + \frac{x_{Z_i^k} v_i \beta \Delta}{f_2+u})}]^{\frac{1}{z}}
\end{aligned}
$$

Also have:

$$
\begin{aligned}
CT_2 &= \prod_{k=1}^{\mathcal{K}} [e(C_A, K_{A_k})^{\frac{1}{z}} \cdot e(C_B, K_{B_k})^{\frac{1}{z}}] \\
&= [e(g^{s_2}, g_2^{\sum_{k=1}^{\mathcal{K}} \alpha_k}) \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{1,i,k})^{-\gamma_k s_2} \\
&\cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{2,i,k})^{-\beta_k s_2} \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, \\
&K_{3,i,k})^{-\zeta_k s_2} \cdot \prod_{k=1}^{\mathcal{K}} \prod_{i=1}^{n} e(g, K_{4,i,k})^{-\varsigma_k s_2} \\
&\cdot e(g,g)^{-\sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} (r_{1,i,k}+r_{2,i,k}) s_1 \Delta}]^{\frac{1}{z}}
\end{aligned}
$$

Finally, we have:

$$
\begin{aligned}
&\frac{C_0}{(CT_1 \cdot CT_2)^z} \qquad (14) \\
&= \frac{M}{e(g,g)^{\frac{(\sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} x_{V_i^k} v_i) \beta \Delta}{f_1+u}} \cdot e(g,g)^{\frac{(\sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n} x_{Z_i^k} v_i) \beta \Delta}{f_2+u}}}
\end{aligned}
$$

Therefore, the message M will be recovered iff $(\overrightarrow{v}, \sum_{k=1}^{\mathcal{K}} \overrightarrow{x_{V^k}}) = 0$ and $(\overrightarrow{v}, \sum_{k=1}^{\mathcal{K}} \overrightarrow{x_{Z^k}}) = 0$, meaning that users' attributes list satisfies the access policy.

5.4 Security Against Attack

A basic requirement of the decentralized ABE scheme is to prevent collusion between users, meaning that any two or more users who are not authorized to decrypt individually can successfully decrypt by combining their keys. In our scheme, GID is introduced to solve this problem as [3], and GID is coupled non-linearly with f_1 and f_2 in the secret keys to resist the attack mentioned by Rahulamathavan *et al.* in [23].

In addition, our scheme is proven to resist the collusion attack mentioned in [29] as follows. Suppose that there are three attribute authorities: AA_1, AA_2, AA_3, which monitor respectively attribute att_1, att_2, att_3. The access policy is specified as W =$\{att_1, att_2, att_3\}$. Consider that two users u_1 and u_2, with attribute sets S_1= $\{att_1, att_2\}$ and S_2= $\{att_2, att_3\}$ respectively, hope to decrypt the ciphertext by collusion.

$$
\begin{aligned}
u_1 &: K_{j,i,1}(u_1), K_{j,i,2}(u_1); (where j = 1,2,3,4) \\
u_2 &: K_{j,i,2}(u_2), K_{j,i,3}(u_2); \\
CT &: C_0 = M \cdot e(g,g_2)^{s_2(\alpha_1+\alpha_2+\alpha_3)}, \\
&\quad C_A, C_B, \{C_{j,i,1}, C_{j,i,2}, C_{j,i,3}\}. \qquad (15)
\end{aligned}
$$

Then we use the secret keys of u_1 and u_2 to decrypt the ciphertext CT. When calculate:

$$
\begin{aligned}
&\prod_{j=1}^{4} [e(C_{j,i,1}, K_{j,i,1}(u_1)) \cdot e(C_{j,i,2}, K_{j,i,2}(u_1)) \\
&\cdot e(C_{j,i,3}, K_{j,i,3}(u_2))] \qquad (16)
\end{aligned}
$$

we find that the collusion is prevented by two special items:

$$
\begin{aligned}
&e(g,g)^{(\frac{x_{V_i^1}}{f_1+u_1} + \frac{x_{V_i^2}}{f_1+u_1} + \frac{x_{V_i^3}}{f_1+u_2}) v_i \beta \Delta} \neq e(g,g)^0 \qquad (17) \\
&e(g,g)^{(\frac{x_{Z_i^1}}{f_2+u_1} + \frac{x_{Z_i^2}}{f_2+u_1} + \frac{x_{Z_i^3}}{f_2+u_2}) v_i \beta \Delta} \neq e(g,g)^0
\end{aligned}
$$

So the message M can not be recovered.

Moreover, the existence of η_k prevents the leakage when the malicious user lets attribute vectors $\overrightarrow{x_{V^k}}$ and $\overrightarrow{x_{Z^k}}$ equal to $\overrightarrow{0}$. If $\eta_k = 0$, the secret value $r_{i,k,1}$ and $r_{i,k,2}$ associated with the attribute will be leaked.

In conclusion, the decentralizing MA-ABE scheme resists strongly attacks from potential malicious users.

6 Security Analysis

Theorem 1. *If the DBDH and DLIN assumption hold in group $\mathbb{G}$, then our decentralizing ABE scheme is selective IND-CPA secure and policy hiding.*

Proof. The proof technique is similar to that of the scheme in [21] except that the corrupted authorities need

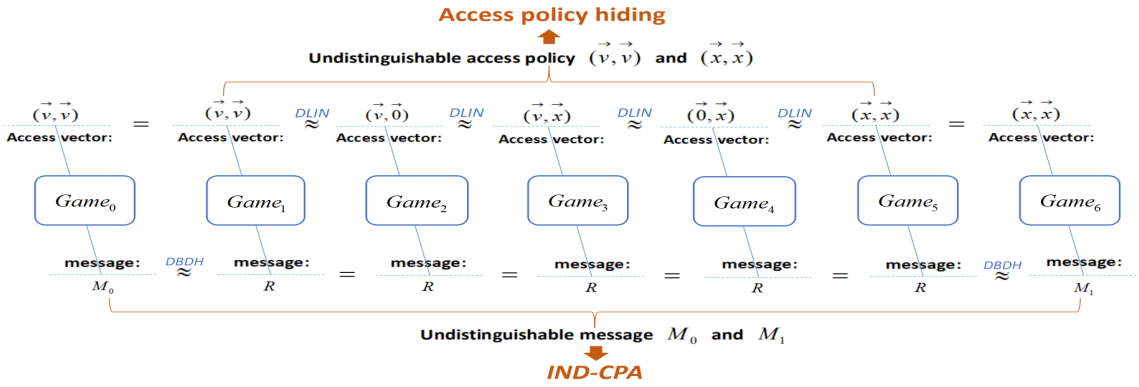


Figure 4: The analysis of the security proof

to be considered in ours. Suppose there is at least one honest center to distribute the private keys in the system. Since the message M is encrypted by an vector that is transformed by the access policy in our scheme. To prove that the policy is hidden, it is required only to prove that the two vectors $\overrightarrow{v}$ and $\overrightarrow{x}$ cannot be distinguished by the adversary, where $\overrightarrow{v}$ and $\overrightarrow{x}$ are corresponding to W_0 and W_1 respectively. □

The following two cases $M_0 = M_1$ and $M_0 \neq M_1$ will be considered. For $M_0 = M_1$, we only prove the property of policy hiding by discussing games from $Game_1$ to $Game_5$ in sequence. For $M_0 \neq M_1$, we need to discuss the whole proof from $Game_0$ to $Game_6$. This specific process is shown in Figure 4.

Firstly, a high level description of each game is given as follows, where $i = 1, ..., n$; $k = 1, ..., \mathcal{K}$.

- $Game_0$: The access policy $(\overrightarrow{v}, \overrightarrow{v})$ is used to encrypt the message M_0. The ciphertext CT_0 is as follows:

$$(M_0 \prod_{k=1}^{\mathcal{K}} Y_k^{s_2}, C_A, C_B, \{U_{1,i,k}^{s_1} T_k^{s_2} V_1^{v_i\beta}, U_{2,i,k}^{s_1} Z_k^{s_2} V_2^{v_i\beta}, W_{1,i,k}^{s_1} M_k^{s_2} X_1^{v_i\beta}, W_{2,i,k}^{s_1} N_k^{s_2} X_2^{v_i\beta}\})$$

- $Game_1$: The access policy $(\overrightarrow{v}, \overrightarrow{v})$ is used to encrypt a random message $R \in \mathbb{G}_{\mathbb{T}}$. The ciphertext CT_1 is as follows:

$$(R', C_A, C_B, \{U_{1,i,k}^{s_1} T_k^{s_2} V_1^{v_i\beta}, U_{2,i,k}^{s_1} Z_k^{s_2} V_2^{v_i\beta}, W_{1,i,k}^{s_1} M_k^{s_2} X_1^{v_i\beta}, W_{2,i,k}^{s_1} N_k^{s_2} X_2^{v_i\beta}\})$$

- $Game_2$: The access policy $(\overrightarrow{v}, \overrightarrow{0})$ is used to encrypt a random message R. The ciphertext CT_2 is as follows:

$$(R', C_A, C_B, \{U_{1,i,k}^{s_1} T_k^{s_2} V_1^{v_i\beta}, U_{2,i,k}^{s_1} Z_k^{s_2} V_2^{v_i\beta}, W_{1,i,k}^{s_1} M_k^{s_2}, W_{2,i,k}^{s_1} N_k^{s_2}\})$$

- $Game_3$: The access policy $(\overrightarrow{v}, \overrightarrow{x})$ is used to encrypt a random message R. The ciphertext CT_3 is as follows:

$$(R', C_A, C_B, \{U_{1,i,k}^{s_1} T_k^{s_2} V_1^{v_i\beta}, U_{2,i,k}^{s_1} Z_k^{s_2} V_2^{v_i\beta}, W_{1,i,k}^{s_1} M_k^{s_2} X_1^{x_i\beta}, W_{2,i,k}^{s_1} N_k^{s_2} X_2^{x_i\beta}\})$$

- $Game_4$: The access policy $(\overrightarrow{0}, \overrightarrow{x})$ is used to encrypt a random message R. The ciphertext CT_4 is as follows:

$$(R', C_A, C_B, \{U_{1,i,k}^{s_1} T_k^{s_2}, U_{2,i,k}^{s_1} Z_k^{s_2}, W_{1,i,k}^{s_1} M_k^{s_2} X_1^{x_i\beta}, W_{2,i,k}^{s_1} N_k^{s_2} X_2^{x_i\beta}\})$$

- $Game_5$: The access policy $(\overrightarrow{x}, \overrightarrow{x})$ is used to encrypt a random message R. The ciphertext CT_5 is as follows:

$$(R', C_A, C_B, \{U_{1,i,k}^{s_1} T_k^{s_2} V_1^{x_i\beta}, U_{2,i,k}^{s_1} Z_k^{s_2} V_2^{x_i\beta}, W_{1,i,k}^{s_1} M_k^{s_2} X_1^{x_i\beta}, W_{2,i,k}^{s_1} N_k^{s_2} X_2^{x_i\beta}\})$$

- $Game_6$: The access policy $(\overrightarrow{x}, \overrightarrow{x})$ is used to encrypt the message $M_1 \in \mathbb{G}_{\mathbb{T}}$. The ciphertext CT_6 is as follows:

$$(M_1 \prod_{k=1}^{\mathcal{K}} Y_k^{s_2}, C_A, C_B, \{U_{1,i,k}^{s_1} T_k^{s_2} V_1^{x_i\beta}, U_{2,i,k}^{s_1} Z_k^{s_2} V_2^{x_i\beta}, W_{1,i,k}^{s_1} M_k^{s_2} X_1^{x_i\beta}, W_{2,i,k}^{s_1} N_k^{s_2} X_2^{x_i\beta}\})$$

6.1 Indistinguishability Between Game$_0$ and Game$_1$

Lemma 1. *For any adversary$\mathcal{A}$, $Game_0$ and $Game_1$ could be distinguished with a non-negligible advantage, then there exists algorithm $\mathcal{B}$ that could solve the DBDH assumption with a non-negligible advantage, i.e.*

$$| Adv_{Game_0}(\lambda) - Adv_{Game_1}(\lambda) | \leq Adv_{\mathcal{B}}^{DBDH}(\lambda) \quad (18)$$

Proof. Let $\overrightarrow{y} = \{g, A = g^a, B = g^b, C = g^c\}$. The challenger $\mathcal{C}$ generates the bilinear group $(e, p, g, \mathbb{G}, \mathbb{G}_{\mathbb{T}})$, then flips an unbiased cion to obtain a bit $\mu \in \{0, 1\}$. If $\mu = 0$, then $\mathcal{C}$ sends $(\overrightarrow{y}, e(g, g)^{abc})$ to $\mathcal{B}$; If $\mu = 1$, then $\mathcal{C}$ sends $(\overrightarrow{y}, R)$ to $\mathcal{B}$, where $R \in_R \mathbb{G}_{\mathbb{T}}$. □

Global setup: $\mathcal{A}$ submits an access vector $(\overrightarrow{v}, \overrightarrow{v})$ corresponding to W_0 and a series of corrupted authorities AA_k^* to $\mathcal{B}$. $\mathcal{B}$ selects randomly μ_1, μ_2, θ_1, θ_2, λ, f_1, f_2, $\Delta \in_R Z_p$ and sets $V_1 = g^{\mu_1}$, $V_2 = g^{\mu_2}$, $X_1 = g^{\theta_1}$, $X_2 = g^{\theta_2}$.

Authority setup: Let I_C be universe authority. There should be three kinds of authority, the corrupted authorities AA_k^*, the honest ones AA_k^{**}, and at least one half-honest authority AA_δ that can only get partial secret key.

1) For corrupted authorities AA_k^*, $\mathcal{B}$ selects randomly α_k, γ_k, β_k, ζ_k, ς_k, η_k, $\{u_{1,i,k}, w_{1,i,k}, u_{2,i,k}, w_{2,i,k}\}_{i=1\ k\in AA_k^*}^{n}$ as secret keys under the condition:$\Delta = \mu_1 u_{2,i,k} - \mu_2 u_{1,i,k} = \theta_1 w_{2,i,k} - \theta_2 w_{1,i,k}$, then calculates $g_2 = g$, $g_1 = g^{\Delta}$, $Y_k = e(g, g)^{\alpha_k}$, for i=1 to n computes:

$$\begin{aligned} U_{1,i,k} &= g^{u_{1,i,k}}, U_{2,i,k} = g^{u_{2,i,k}} \\ W_{1,i,k} &= g^{w_{1,i,k}}, W_{2,i,k} = g^{w_{2,i,k}} \\ T_k &= g^{\gamma_k}, Z_k = g^{\beta_k}, M_k = g^{\zeta_k}, N_k = g^{\varsigma_k} \end{aligned}$$

as the attribute public keys. $\mathcal{B}$ sends authorities AA_k^*'s secret keys $SK_k = (\alpha_k, \gamma_k, \beta_k, \zeta_k, \varsigma_k, \eta_k, \{u_{1,i,k}, u_{2,i,k}, w_{1,i,k}, w_{2,i,k}\}_{i=1}^{n})_{k\in AA_k^*}$ and public keys $PK_k = (g_1, Y_k, T_k, Z_k, M_k, N_k, \{U_{1,i,k}, U_{2,i,k}, W_{1,i,k}, W_{2,i,k}\}_{i=1}^{n})_{k\in AA_k^*}$ to $\mathcal{A}$.

2) For the honest authorities AA_k^{**}, $\mathcal{B}$ selects randomly α_k, γ_k, β_k, ζ_k, ς_k, η_k,$\{u_{1,i,k}, w_{1,i,k}, u_{2,i,k}, w_{2,i,k}\}_{i=1 k\in AA_k^{**}}^{n}$ as secret keys under the condition: $\Delta = \mu_1 u_{2,i,k} - \mu_2 u_{1,i,k} = \theta_1 w_{2,i,k} - \theta_2 w_{1,i,k}$, then lets $g_2 = g^b$, and calculates $g_1 = g^{\Delta}$, $Y_k = e(g,g)^{b\alpha_k}$, $T_k = g^{\gamma_k}$, $Z_k = g^{\beta_k}$, $M_k = g^{\zeta_k}$, $N_k = g^{\varsigma_k}$ as public keys. $\mathcal{B}$ calculates the attribute public keys for $j = 1, 2$ as follows:

$$U_{j,i,k} = \begin{cases} g^{u_{j,i,k}} & v_i \in (\overrightarrow{v}, \overrightarrow{v}) \\ g^{bu_{j,i,k}} & v_i \notin (\overrightarrow{v}, \overrightarrow{v}) \end{cases} \quad (19)$$

$$W_{j,i,k} = \begin{cases} g^{w_{j,i,k}} & v_i \in (\overrightarrow{v}, \overrightarrow{v}) \\ g^{bw_{j,i,k}} & v_i \notin (\overrightarrow{v}, \overrightarrow{v}) \end{cases}$$

$\mathcal{B}$ sends honest authority AA_k^{**}'s public keys $PK_k = (g_1, \{Y_k, T_k, Z_k, M_k, N_k\}, \{U_{1,i,k}, U_{2,i,k}, W_{1,i,k}, W_{2,i,k}\}_{i=1}^{n})_{k\in AA_k^{**}}$ to $\mathcal{A}$.

3) For the half-honest authority AA_δ, it is same as the second case except that $\mathcal{B}$ calculates g_1=g^{Δ}, Y_k=$e(g, g)^{ab} \cdot \prod_{k\in AA_k^*} e(g,g)^{-\alpha_k} \cdot \prod_{k\in AA_k^{**}} e(g, g)^{-b\alpha_k}$.

Phase 1: $\mathcal{A}$ submits the attributes list S and GID for secret keys queries. $\mathcal{B}$ chooses random $u^{'} \in Z_p$ for H(GID). $\mathcal{A}$ can query polynomially. Consider a query with two vectors $\overrightarrow{x_{V^k}} = (x_{V_1^k}, x_{V_2^k}, ..., x_{V_n^k})$ and $\overrightarrow{x_{Z^k}} = (x_{Z_1^k}, x_{Z_2^k}, ..., x_{Z_n^k})$, which is related to attributes in $S^k = U_k \bigcap S$. $\mathcal{A}$ can query the secret keys as long as $(\overrightarrow{v}, \overrightarrow{x_{V^k}}) = (\overrightarrow{v}, \overrightarrow{x_{Z^k}}) \neq 0$.

1) For corrupted authorities AA_k^*: $\mathcal{B}$ computes secret keys $SK_{L_k^*}$ for attributes in $S^{k^*} = U_k^* \bigcap S$ to $u^{'}$, where U_k^* is AA_k^*'s attributes set.

2) For the honest authorities AA_k^{**}: $\mathcal{B}$ picks random exponents $\{r_{1,i,k}, r_{2,i,k}\}_{i=1,k\in AA_k^{**}}^{n} \in_R Z_p$, then $\mathcal{B}$ computes

$$\begin{aligned} K_{1,i,k} &= g^{-\mu_2 r_{1,i,k}} \cdot U_{2,i,k}^{\frac{x_{V_i^k}}{u^{'}+f_1}+\eta_k} \quad (20) \\ K_{2,i,k} &= g^{\mu_1 r_{1,i,k}} \cdot U_{1,i,k}^{-(\frac{x_{V_i^k}}{u^{'}+f_1}+\eta_k)} \\ K_{3,i,k} &= g^{-\theta_2 r_{2,i,k}} \cdot W_{2,i,k}^{\frac{x_{Z_i^k}}{u^{'}+f_2}-\eta_k} \\ K_{4,i,k} &= g^{\theta_1 r_{2,i,k}} \cdot W_{1,i,k}^{-(\frac{x_{Z_i^k}}{u^{'}+f_2}-\eta_k)} \end{aligned}$$

Then K_{A_k}, K_{B_k} is calculated as:

$$\begin{aligned} K_{B_k} &= \prod_{i=1}^{n} g^{-(r_{1,i,k}+r_{2,i,k})\Delta} \quad (21) \\ K_{A_k} &= B^{\alpha_k} \cdot \prod_{i=1}^{n} K_{1,i,k}^{-\gamma_k} \cdot K_{2,i,k}^{-\beta_k} \\ &\quad \cdot K_{3,i,k}^{-\zeta_k} \cdot K_{4,i,k}^{-\varsigma_k} \end{aligned}$$

3) For the half-honest authority AA_δ: $\mathcal{B}$ selects random $\{r_{1,i,\delta}, r_{2,i,\delta}\}_{i=1}^{n} \in_R Z_p$, then $\mathcal{B}$ computes K_{A_δ} as follows:

$$\begin{aligned} K_{A_\delta} =& B^{-\lambda} \prod_{i=1}^{n} (K_{1,i,\delta}^{-\gamma_k} \cdot K_{2,i,\delta}^{-\beta_k} \cdot K_{3,i,\delta}^{-\zeta_k} \\ & \cdot K_{4,i,\delta}^{-\varsigma_k}) \cdot \prod_{k\in A_k^*} g^{-\alpha_k} \cdot \prod_{k\notin A_k^*} B^{-\alpha_k} \end{aligned}$$

We claim that K_{A_δ} is a valid secret key as follows:

$$\begin{aligned} K_{A_\delta} =& B^{-\lambda} \prod_{i=1}^{n} (K_{1,i,\delta}^{-\gamma_k} K_{2,i,\delta}^{-\beta_k} K_{3,i,\delta}^{-\zeta_k} \\ & K_{4,i,\delta}^{-\varsigma_k}) \cdot \prod_{k\in A_k^*} g^{-\alpha_k} \cdot \prod_{k\notin A_k^*} B^{-\alpha_k} \\ =& g^{ab-(\sum_{k\in A_k^*}\alpha_k + \sum_{k\notin A_k^*} b\alpha_k)} \prod_{i=1}^{n} [\\ & g^{\mu_2(r_{1,i,\delta}-b)\gamma_k} \cdot g^{-\mu_1(r_{1,i,\delta}-b)\beta_k} \\ & \cdot g^{\theta_2(r_{2,i,\delta}-b)\zeta_k} \cdot g^{-\theta_1(r_{2,i,\delta}-b)\varsigma_k} \\ & \cdot U_{2,i,\delta}^{-(\frac{x_{V_i^k}}{u^{'}+f_1}+\eta_k)\gamma_k} \cdot U_{1,i,\delta}^{(\frac{x_{V_i^k}}{u^{'}+f_1}+\eta_k)\beta_k} \\ & \cdot W_{2,i,\delta}^{-(\frac{x_{Z_i^k}}{u^{'}+f_2}-\eta_k)\zeta_k} \cdot W_{1,i,\delta}^{(\frac{x_{Z_i^k}}{u^{'}+f_2}-\eta_k)\varsigma_k}] \\ =& g^{ab-(\sum_{k\in A_k^*}\alpha_k + \sum_{k\notin A_k^*} b\alpha_k)} \cdot \prod_{i=1}^{n} (K_{1,i,\delta}^{'-\gamma_k} \\ & \cdot K_{2,i,\delta}^{'-\beta_k} \cdot K_{3,i,\delta}^{'-\zeta_k} \cdot K_{4,i,\delta}^{'-\varsigma_k}) \end{aligned}$$

Where lets $r_{1,i,\delta}^{'} = r_{1,i,\delta} - b$, $r_{2,i,\delta}^{'} = r_{2,i,\delta} - b$, and implicitly sets: $\mu_2\gamma_k - \mu_1\beta_k + \theta_2\zeta_k - \theta_1\varsigma_k = a + \lambda$. Note that $K_{1,i,\delta}$,$K_{2,i,\delta}$,$K_{3,i,\delta}$,$K_{4,i,\delta}$ and K_{B_δ} is same as Equations (21) and (22).

$\mathcal{B}$ gives $\mathcal{A}$ the secret keys $SK_{GID,k,i} = (K_{A_k}, K_{B_k}, \{K_{1,i,k}, K_{2,i,k}, K_{3,i,k}, K_{4,i,k}\}_{i=1}^{n})_{k\in AA_k^*, k\in AA_k^{**}, k\in AA_\delta}$ for the queried attributes set S.

Challenge: $\mathcal{A}$ submits two equal length messages M_0 and M_1 to $\mathcal{B}$. $\mathcal{B}$ selects a random bit $\xi \in_R \{0,1\}$ and runs *Encryption*(PK_k, M_ξ). $\mathcal{B}$ selects randomly $s_1^{'}, \beta^{'} \in Z_p$, and implicitly sets $s_1 = s_1^{'}, s_2 = c, \beta = \beta^{'}$. For i from 1 to n, $\mathcal{B}$ computes as:

$$C_0 = M_\xi \cdot Z;\ C_A = g^c;\ C_B = (g^{\triangle})^{s_1^{'}}; \tag{22}$$

$$C_{1,i,k} = U_{1,i,k}^{s_1^{'}}(g^{\gamma_k})^c g^{\mu_1 v_i \beta^{'}}, C_{2,i,k} = U_{2,i,k}^{s_1^{'}}(g^{\beta_k})^c g^{\mu_2 v_i \beta^{'}}$$

$$C_{3,i,k} = W_{1,i,k}^{s_1^{'}}(g^{\zeta_k})^c g^{\theta_1 v_i \beta^{'}}, C_{4,i,k} = W_{2,i,k}^{s_1^{'}}(g^{\varsigma_k})^c g^{\theta_2 v_i \beta^{'}}$$

$\mathcal{B}$ sends the ciphertext $CT = (\ C_0, C_A, C_B, \{C_{1,i,k}, C_{2,i,k}, C_{3,i,k}, C_{4,i,k}\}_{i=1}^{n}{}_{k=1}^{\mathcal{K}})$ to $\mathcal{A}$.

If $\mu = 0$, then $Z = e(g,g)^{abc}$, we can show CT is a valid ciphertext of message M_ξ by computing

$$\prod_{k\in I_C} Y_k{}^c = \prod_{k\in A_k^*} e(g,g)^{\alpha_k} \prod_{k\in A_k^{**}} e(g,g)^{b\alpha_k} [e(g,g)^{abc} \cdot \prod_{k\in A_k^*} e(g,g)^{-\alpha_k} \cdot \prod_{k\in A_k^{**}} e(g,g)^{-b\alpha_k}] = Z$$

Phase 2: Phase 1 is repeated.

Guess: Finally $\mathcal{A}$ returns his guess $\xi^{'}$ on ξ. If $\xi^{'} = \xi$, $\mathcal{B}$ returns his guess $\mu^{'} = 0$ on μ, otherwise, $\mathcal{B}$ returns his guess $\mu^{'} = 1$ on μ.

$\mathcal{A}$ can get nothing about his guess on ξ when $\mu = 1$ since the input of Z is a random number z. Therefore, $\mathcal{A}$ cannot distinguish ξ with non-negligible advantage, so $\Pr[\xi^{'} \neq \xi | \mu = 1] = \frac{1}{2}$, $\mathcal{B}$ returns his guess $\mu^{'} = 1$ when $\xi^{'} \neq \xi$, thus we have $\Pr[\mu^{'} = \mu | \mu = 1] = \frac{1}{2}$.

If $\mu = 0$, according to the definition of DBDH complexity assumption, the advantage of adversary $\mathcal{A}$ in outputing $\xi^{'} = \xi$ is at least ϵ. Therefore, we have $\Pr[\xi^{'} = \xi | \mu = 0] \geq \frac{1}{2} + \epsilon$. When $\xi^{'} = \xi$, $\mathcal{B}$ returns $\mu^{'} = 0$ on μ, so we have $\Pr[\mu^{'} = \mu | \mu = 0] \geq \frac{1}{2} + \epsilon$.

In conclusion, $\mathcal{B}$'s advantage to break the DBDH assumption is $|\frac{1}{2}\Pr[\mu^{'} = \mu | \mu = 0] + \frac{1}{2}\Pr[\mu^{'} = \mu | \mu = 1] - \frac{1}{2}| \geq \frac{\epsilon}{2}$. Hence, if adversary $\mathcal{A}$ can distinguish these two games, $\mathcal{B}$ can solve the DBDH problem.

6.2 Indistinguishability Between Game_1 and Game_2

Lemma 2. *For any adversary $\mathcal{A}$, $Game_1$ and $Game_2$ could be distinguished with a non-negligible advantage, then there exists algorithm $\mathcal{B}$ that could solve the DLIN assumption with a non-negligible advantage, i.e.*

$$| Adv_{Game_1}(\lambda) - Adv_{Game_2}(\lambda) | \leq Adv_{\mathcal{B}}^{DLIN}(\lambda) \tag{23}$$

Proof. Let $\overrightarrow{y} = \{g, Z_1 = g^{z_1}, Z_2 = g^{z_2}, Z_3 = g^{z_2 z_4}, Z_4 = g^{z_3 + z_4}\}$. The challenger $\mathcal{C}$ generates the bilinear group (e, p, g, $\mathbb{G}$, $\mathbb{G}_\mathbb{T}$), and flips an unbiased cion with $\{0,1\}$ to obtains a bit μ. If $\mu = 0$, $\mathcal{C}$ sends ($\overrightarrow{y}$, $g^{z_1 z_3}$) to $\mathcal{B}$; If $\mu = 1$, then $\mathcal{C}$ sends ($\overrightarrow{y}$, R) to $\mathcal{B}$, where $R \in_R \mathbb{G}_\mathbb{T}$. □

Global setup: $\mathcal{A}$ submits two access vectors $(\overrightarrow{v}, \overrightarrow{v})$ and $(\overrightarrow{v}, \overrightarrow{0})$ which is corresponding to W_0 and W_1, and sends a list of corrupted authorities AA_k^* to $\mathcal{B}$. Then $\mathcal{B}$ selects randomly μ_1, μ_2, θ_1, θ_2, f_1, f_2, $\Delta \in_R Z_p$, and sets $V_1 = g^{\mu_1}$, $V_2 = g^{\mu_2}$, $X_1 = g^{\theta_1}$, $X_2 = g^{\theta_2}$.

Authorities setup: Let I_C be universe authority. There should be three kinds of authorities, the corrupted authorities AA_k^*, the honest ones AA_k^{**} and at least one half-honest authority AA_δ for which $\mathcal{A}$ can only get parts of the secret keys.

1) For corrupted authorities AA_k^*: $\mathcal{B}$ is same as the algorithm of corruption authorities during *Authorities Setup* in the proof of *lemma 1.*

2) For the honest authorities AA_k^{**}: $\mathcal{B}$ selects randomly $\alpha_k, \gamma_k, \beta_k, \zeta_k, \varsigma_k, \eta_k, \{u_{1,i,k}, w_{1,i,k}, u_{2,i,k}, w_{2,i,k}\}_{i=1}^{n}{}_{k \in AA_k^{**}}$ as secret keys under the condition: $\Delta = \mu_1 u_{2,i,k} - \mu_2 u_{1,i,k} = \theta_1 w_{2,i,k} - \theta_2 w_{1,i,k}$, then lets $g_2 = g^{z_2}$, and calculates $g_1 = g^{\Delta}$, $Y_k = e(g,g)^{z_2 \alpha_k}$, $T_k = g^{\gamma_k}$, $Z_k = g^{\beta_k}$, $M_k = g^{\zeta_k}$, $N_k = g^{\varsigma_k}$ as public keys. $\mathcal{B}$ calculates the attribute public keys for $j = 1$ to 2 as follows:

$$U_{j,i,k} = \begin{cases} g^{u_{j,i,k}} & v_i \in (\overrightarrow{v}, \overrightarrow{v}) or (\overrightarrow{v}, \overrightarrow{0}) \\ g^{z_1 u_{j,i,k}} & v_i \notin (\overrightarrow{v}, \overrightarrow{v}) or (\overrightarrow{v}, \overrightarrow{0}) \end{cases}$$

$$W_{j,i,k} = \begin{cases} g^{w_{j,i,k}} & v_i \in (\overrightarrow{v}, \overrightarrow{v}) or (\overrightarrow{v}, \overrightarrow{0}) \\ g^{z_1 w_{j,i,k}} & v_i \notin (\overrightarrow{v}, \overrightarrow{v}) or (\overrightarrow{v}, \overrightarrow{0}) \end{cases}$$

$\mathcal{B}$ sends honest authority AA_k^{**}'s public keys $PK_k = (g_1, \{Y_k, T_k, Z_k, M_k, N_k\}, \{U_{1,i,k}, U_{2,i,k}, W_{1,i,k}, W_{2,i,k}\}_{i=1}^{n})_{k \in AA_k^{**}}$ to $\mathcal{A}$.

3) For the half-honest authority AA_δ, it is same as the second case except that $\mathcal{B}$ calculates $g_1 = g^{\Delta}$, $Y_k = e(g,g)^{z_2} \cdot \prod_{k \in AA_k^*} e(g,g)^{-\alpha_k} \cdot \prod_{k \in AA_k^{**}} e(g,g)^{-z_2 \alpha_k}$.

Phase 1: $\mathcal{A}$ submits the attributes list S and GID for secret keys queries. $\mathcal{B}$ chooses random $u^{'} \in Z_p$ for H(GID). $\mathcal{A}$ can query polynomially. Consider a query with two vectors $\overrightarrow{x_{V^k}} = (x_{V_1^k}, x_{V_2^k}, ..., x_{V_n^k})$ and $\overrightarrow{x_{Z^k}} = (x_{Z_1^k}, x_{Z_2^k}, ..., x_{Z_n^k})$, which is related to attributes in $L_k = U_k \bigcap L$. $\mathcal{A}$ can query the secret keys as long as $(\overrightarrow{v}, \overrightarrow{x_{V^k}}) = (\overrightarrow{v}, \overrightarrow{x_{Z^k}}) \neq 0$.

1) For corrupted authorities AA_k^*: $\mathcal{B}$ computes secret keys $SK_{L_k^*}$ for attributes in $S^{k^*} = U_k^* \bigcap S$ to $u^{'}$, where U_k^* is the attribute set of the authorities AA_k^*.

2) For the honest authorities AA_k^{**}: $\mathcal{B}$ is same as the algorithm of honest authorities during *Phase 1* in the proof of *lemma 1.*

3) For the half-honest authority AA_δ: $\mathcal{B}$ selects random $\{r_{1,i,\delta}, r_{2,i,\delta}\}_{i=1}^{n} \in_R Z_p$, then $\mathcal{B}$ computes K_{A_δ} as fol-

Table 3: The comparison of storage costs at different phases

Scheme	Authority Setup	Encryption	KeyGen	TransKeyGen
[7]	$I(\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_\mathbb{T}\rvert)$	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_\mathbb{T}\rvert$	$p_i N_S \lvert\mathbb{G}\rvert$	–
[30]	$(I+\mathcal{K})\lvert\mathbb{G}\rvert + I\lvert\mathbb{G}_\mathbb{T}\rvert$	$3\lvert\mathbb{G}\rvert + 3N_W\lvert\mathbb{G}_\mathbb{T}\rvert$	$2N_S\lvert\mathbb{G}\rvert$	–
[1]	$(I+\mathcal{K})\lvert\mathbb{G}\rvert + I\lvert\mathbb{G}_\mathbb{T}\rvert$	$4\lvert\mathbb{G}\rvert + 3N_W\lvert\mathbb{G}_\mathbb{T}\rvert$	$2N_S\lvert\mathbb{G}\rvert$	$2N_S\lvert\mathbb{G}\rvert + 3\lvert\mathbb{G}\rvert$
[17]	$2I\lvert\mathbb{G}\rvert + I\lvert\mathbb{G}_\mathbb{T}\rvert$	$(1+N_W)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_\mathbb{T}\rvert$	$N_S\lvert\mathbb{G}\rvert$	–
[25]	$(I+\mathcal{K})\lvert\mathbb{G}\rvert + I\lvert\mathbb{G}_\mathbb{T}\rvert$	$(2N_W+1)\lvert G\rvert + (N_W+1)\lvert\mathbb{G}_\mathbb{T}\rvert$	$2N_S\lvert\mathbb{G}\rvert$	$N_S(\lvert\mathbb{G}\rvert + \lvert\mathcal{O}(\mathcal{H})\rvert)$
Ours	$(4+5\mathcal{K})\lvert\mathbb{G}\rvert + \mathcal{K}\lvert\mathbb{G}_\mathbb{T}\rvert$	$(2+4\rho N_W)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_\mathbb{T}\rvert$	$(2+4\rho N_W)\lvert\mathbb{G}\rvert$	$(2+4\rho N_W)\lvert\mathbb{G}\rvert$

[1] $\lvert\mathbb{G}\rvert$: the size of one element in the group $\mathbb{G}$. $\lvert\mathbb{G}_\mathbb{T}\rvert$: the size of one element in the group $\mathbb{G}_\mathbb{T}$.
[2] N_W: the number of attributes in the access policy. N_S: The number of attributes in user's attribute set. I: The number of attributes in the system.
[3] $\lvert\mathcal{O}(\mathcal{H})\rvert$: The size of a hash function. $\rho \in (0,1)$: The coefficient associated with the number of wildcards.
[4] $\mathcal{K}$: The number of the attribute authority.

lows:

$$K_{A_\delta} = Z_2^{-\lambda'} \cdot \prod_{i=1}^{n} (K_{1,i,\delta}{}^{-\gamma_k} K_{2,i,\delta}{}^{-\beta_k} K_{3,i,\delta}{}^{-\zeta_k} K_{4,i,\delta}{}^{-\varsigma_k}) \prod_{k\in A_k^*} g^{-\alpha_k} \prod_{k\notin A_k^*} Z_2^{-\alpha_k} \quad (24)$$

We claim that K_{A_δ} is a valid secret key as follows:

$$\begin{aligned}
K_{A_\delta} =& Z_2^{-\lambda'} \prod_{i=1}^{n} (K_{1,i,\delta}{}^{-\gamma_k} K_{2,i,\delta}{}^{-\beta_k} K_{3,i,\delta}{}^{-\zeta_k} K_{4,i,\delta}{}^{-\varsigma_k}) \cdot \prod_{k\in A_k^*} g^{-\alpha_k} \cdot \prod_{k\notin A_k^*} Z_2^{-\alpha_k} \\
=& g^{z_2 - (\sum_{k\in A_k^*}\alpha_k + \sum_{k\notin A_k^*} z_2\alpha_k)} \\
& \cdot \prod_{i=1}^{n} g^{\mu_2(r_{1,i,\delta}-z_2)\gamma_k} \cdot g^{-\mu_1(r_{1,i,\delta}-z_2)\beta_k} \\
& \cdot g^{\theta_2(r_{2,i,\delta}-z_2)\zeta_k} \cdot g^{-\theta_1(r_{2,i,\delta}-z_2)\varsigma_k} \\
& \cdot U_{2,i,\delta}{}^{-(\frac{x_{V_i^k}}{u'+f_1}+\eta_k)\gamma_k} \cdot U_{1,i,\delta}{}^{(\frac{x_{V_i^k}}{u'+f_1}+\eta_k)\beta_k} \\
& \cdot W_{2,i,\delta}{}^{-(\frac{x_{V_i^k}}{u'+f_2}-\eta_k)\zeta_k} \cdot W_{1,i,\delta}{}^{(\frac{x_{V_i^k}}{u'+f_2}-\eta_k)\varsigma_k}] \\
=& g^{z_2 - (\sum_{k\in A_k^*}\alpha_k + \sum_{k\notin A_k^*} z_2\alpha_k)} \cdot \prod_{i=1}^{n} (K'_{1,i,\delta}{}^{-\gamma_k} \\
& \cdot K'_{2,i,\delta}{}^{-\beta_k} \cdot K'_{3,i,\delta}{}^{-\zeta_k} \cdot K'_{4,i,\delta}{}^{-\varsigma_k})
\end{aligned}$$

Where lets $r'_{1,i,\delta} = r_{1,i,\delta} - z_2$, $r'_{2,i,\delta} = r_{2,i,\delta} - z_2$, $\lambda' = \lambda - 1$, and implicitly sets: $\mu_2\gamma_k - \mu_1\beta_k + \theta_2\zeta_k - \theta_1\varsigma_k = \lambda$. Note that $K_{1,i,\delta}$,$K_{2,i,\delta}$,$K_{3,i,\delta}$,$K_{4,i,\delta}$ and K_{B_δ} is same as *lemma 1*.

Finally, $\mathcal{B}$ gives $\mathcal{A}$ the secret keys $SK_{GID,k,i} = (K_{A_k}, K_{B_k}, \{K_{1,i,k}, K_{2,i,k}, K_{3,i,k}, K_{4,i,k}\}_{i=1}^{n}{}_{k\in A, A_k^*, k\in AA_k^{**}, k\in AA_\delta})$ for the queried attribute set L.

Challenge: $\mathcal{A}$ submits two equal length messages M_0 and M_1. $\mathcal{B}$ selects a random bit $\xi \in_R \{0,1\}$ and runs $Encryption(PK_k, W_\xi, M_\xi)$. $\mathcal{B}$ selects random s'_1, $\beta' \in Z_p$, and implicitly sets $s_1 = s'_1, s_2 = z_3 + z_4, \beta = \beta'$.

For i from 1 to n, $\mathcal{B}$ computes as:

$$\begin{aligned}
C_{1,i,k} =& g^{u_{1,i,k}z_1 s'_1}(g^{\gamma_k})^{z_3+z_4} g^{\mu_1 v_i \beta'} = U_{1,i,k}^{s_1} T_k^{s_2} V_1^{v_i\beta'} \\
C_{2,i,k} =& g^{u_{2,i,k}z_1 s'_1}(g^{\beta_k})^{z_3+z_4} g^{\mu_2 v_i \alpha'} = U_{2,i,k}^{s_1} Z_k^{s_2} V_2^{v_i\beta'}
\end{aligned}$$

It implies $s'_1 = z_3$, if $\mu = 0$, then $Z = g^{z_1 z_3}$, then $\mathcal{B}$ is simulating $Game_1$:

$$\begin{aligned}
C_{3,i,k} =& g^{w_{1,i,k}z_1 s'_1}(g^{\zeta_k})^{z_3+z_4} g^{\theta_1 v_i \beta'} = W_{1,i,k}^{s_1} M_k^{s_2} X_1^{v_i\beta'} \\
C_{4,i,k} =& g^{w_{2,i,k}z_1 s'_1}(g^{\varsigma_k})^{z_3+z_4} g^{\theta_2 v_i \beta'} = W_{2,i,k}^{s_1} N_k^{s_2} X_2^{v_i\beta'}
\end{aligned}$$

It implies $s'_1 = z_3 \cdot z (z \in_R \mathbb{G})$, if $\mu = 1$, then $Z = g^{z_1 z_3 \cdot z} = R$, then $\mathcal{B}$ is simulating $Game_2$:

$$\begin{aligned}
C_{3,i,k} &= g^{w_{1,i,k}z_1 s'_1}(g^{\zeta_k})^{z_3+z_4} g^{\theta_1 v_i \beta'} = W_{1,i,k}^{s_1} M_k^{s_2} \\
C_{4,i,k} &= g^{w_{2,i,k}z_1 s'_1}(g^{\varsigma_k})^{z_3+z_4} g^{\theta_2 v_i \beta'} = W_{2,i,k}^{s_1} N_k^{s_2}
\end{aligned}$$

Finally, $\mathcal{B}$ calculates:

$$\begin{aligned}
C_0 =& M_\xi \cdot \prod_{k\in I_C} Y_k^{s_2} \quad (25) \\
=& M_\xi \cdot e(g,g)^{(z_3+z_4)z_2} \cdot \prod_{k\in A_k^*} e(g,g)^{(z_3+z_4)\alpha_k} \cdot \\
& \prod_{k\in A_k^{**}} e(g,g)^{(z_3+z_4)z_2\alpha_k} \prod_{k\in A_k^*} e(g,g)^{-(z_3+z_4)\alpha_k} \\
& \cdot \prod_{k\in A_k^{**}} e(g,g)^{-(z_3+z_4)z_2\alpha_k} \\
=& M_\xi \cdot e(g,g)^{(z_3+z_4)z_2} \\
C_A =& g^{z_3+z_4}, C_B = (g^{\triangle})^{z_3}
\end{aligned}$$

$\mathcal{B}$ sends the ciphertext $CT = (C_0, C_A, C_B, \{C_{1,i,k}, C_{2,i,k}, C_{3,i,k}, C_{4,i,k}\}_{i\in[1,n], k\in AA_k^{**}, k\in AA_\delta})$ to $\mathcal{A}$.

Phase 2: Phase 1 is repeated.

Guess: Finally $\mathcal{A}$ returns his guess ξ' on ξ. If $\xi' = \xi$, $\mathcal{B}$ returns his guess $\mu' = 0$ on μ, otherwise, $\mathcal{B}$ returns his guess $\mu' = 1$ on μ.

Similarly, $\mathcal{B}$'s advantage to break the DLIN assumption is $|\frac{1}{2}\Pr[\mu' = \mu|\mu = 0] + \frac{1}{2}\Pr[\mu' = \mu|\mu = 1] - \frac{1}{2}| \geq \frac{\epsilon}{2}$. Hence, if $\mathcal{A}$ can distinguish these two games, $\mathcal{B}$ can solve the DLIN assumption.

Table 4: The comparison of computation costs at different phases

Scheme	Encryption	User.Decryption	Out.Decryption
[7]	$3E_{\mathbb{G}}$	$2\hat{e}$	–
[30]	$2\hat{e}+(1+2N_W)E_{\mathbb{G}_{\mathbb{T}}}+3N_W E_{\mathbb{G}}$	$(1+2N_S)\hat{e}+N_S E_{\mathbb{G}_{\mathbb{T}}}$	–
[1]	$5E_{\mathbb{G}}+E_{\mathbb{G}_{\mathbb{T}}}+3\mathcal{O}(\mathcal{H})$	$E_{\mathbb{G}_{\mathbb{T}}}+3\mathcal{O}(\mathcal{H})$	$3N_S\hat{e}+E_{\mathbb{G}_{\mathbb{T}}}$
[17]	$\hat{e}+(1+N_W)E_{\mathbb{G}}+E_{\mathbb{G}_{\mathbb{T}}}$	$2\hat{e}+N_S E_{\mathbb{G}}$	–
[25]	$E_{\mathbb{G}_{\mathbb{T}}}+E_{\mathbb{G}}$	$E_{\mathbb{G}_{\mathbb{T}}}$	$2N_S\hat{e}+N_S(E_{\mathbb{G}}+E_{\mathbb{G}_{\mathbb{T}}})$
Ours	$\hat{e}+4\rho N_W E_{\mathbb{G}}+E_{\mathbb{G}_{\mathbb{T}}}$	$E_{\mathbb{G}_{\mathbb{T}}}$	$(2+4\rho N_S)\hat{e}$

[1] $E_{\mathbb{G}}$: The time of an exponential operation in the group $\mathbb{G}$. $E_{\mathbb{G}_{\mathbb{T}}}$: The time of an exponential operation in the group $\mathbb{G}_{\mathbb{T}}$.
[2] $\hat{e}$: The time of computing a pairing function e. $\mathcal{O}(\mathcal{H})$: The time of computing a hash function.

6.3 Indistinguishability Between $Game_2$ and $Game_3$

Lemma 3. *For any adversary $\mathcal{A}$, $Game_2$ and $Game_3$ could be distinguished with a non-negligible advantage, then there exists algorithm $\mathcal{B}$ that could solve the DLIN assumption with a non-negligible advantage,* i.e.

$$| Adv_{Game_2}(\lambda) - Adv_{Game_3}(\lambda) | \leq Adv_{\mathcal{B}}^{DLIN}(\lambda)$$

Proof. Except for *phase 1*, the rest is the same as the above proof in Lemma 2. There are two cases in *phase 1*:

- $(\overrightarrow{v}, \overrightarrow{x_{V^k}}) = (\overrightarrow{v}, \overrightarrow{x_{Z^k}}) = 0 \pmod p$.
- $(\overrightarrow{v}, \overrightarrow{x_{V^k}}) = c_v \neq 0$, $(\overrightarrow{v}, \overrightarrow{x_{Z^k}}) = c_x \neq 0$.

□

Similarly, we can prove the indistinguishability between $Game_3$ and $Game_4$ in the similar way as for that of $Game_2$ and $Game_3$. The proof of indistinguishability between $Game_4$ and $Game_5$ is similar to that of $Game_1$ and $Game_2$. The proof of indistinguishability between $Game_5$ and $Game_6$ is similar to that of $Game_0$ and $Game_1$.

7 Performance Analysis

In this section, we evaluate the storage costs and computation costs of our scheme. For this purpose, we introduce the size of the system's public keys, the ciphertext, the user's secret keys and the transform-keys. In addition, we consider the computation costs related to execution of Encryption, User.Decryption and Out.Decryption Algorithm, which those algorithms are performed by DO, DU and CP, respectively.

To compare the performance of those schemes more intuitively, we give here an empirical comparison of storage costs and computation costs in ours, and the results with the latest the work of Belguith *et al.* [1], the work of Michdevsky *et al.* [17] and the work of Shao *et al.* [25]. We conduct our experiments on a Windows machine with 3.40 GHz Intel(R) Core(TM) i3-3240 CPU and 4 GB RAM. The code uses Pairing Based Cryptography library to achieve the access control scheme, which supports pairing operation. Type A pairings are used in the simulation, which are constructed on the curve over the field for some prime q. The pairing is symmetric, where the order of groups is 160 bits, the base field size is 512 bits. All that the length of an element in each group G and the target group G_T is set to 512 bits.

7.1 Storage Costs

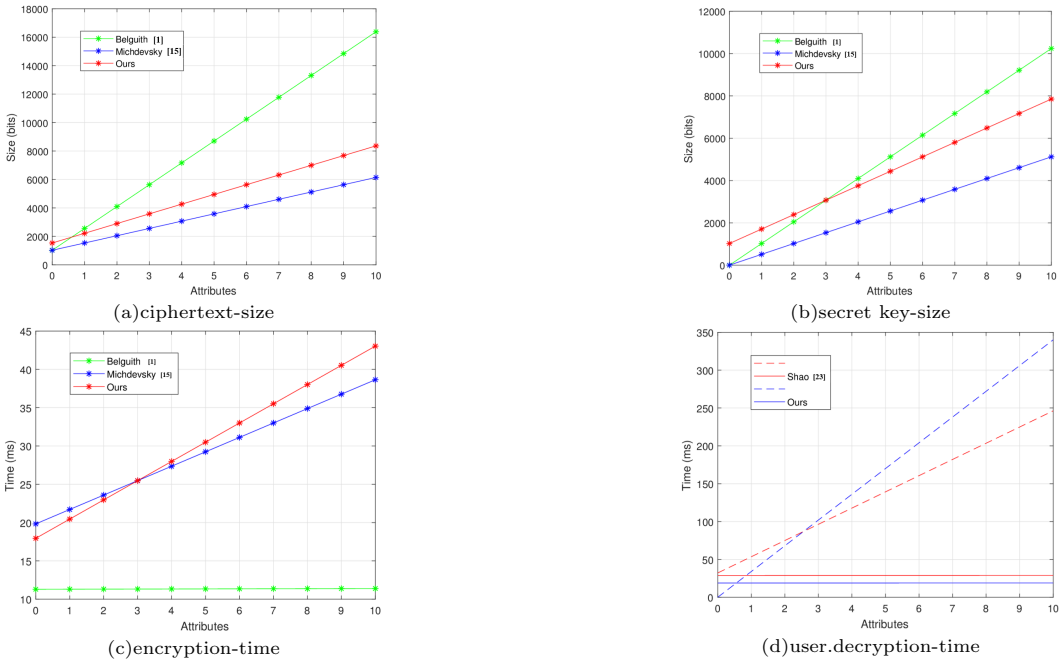
Based on Table 1, we find that our scheme achieve the optimal compromise between policy-hiding fully and the efficiency on the user side. Compared with [1, 21, 25] and [17], our scheme is more flexible in multi-authority environments. We make a comparison between latest closely MA-ABE schemes and our scheme with regard to the size of public keys, ciphertext, secret keys and transform keys in Table 3. From Table 3, we can know that the size of the ciphertext significantly is shorter than that in [1, 30] and [25] when $\rho < 0.8$, and the size of the secret key is shorter than that in [1,7,30] and [25] when $\rho < 0.5$, which depends on the number of the wildcards instead that of the attributes.

In addition, the results in Figure 5(a) and Fifure 5(b) reveal the storage costs of the ciphertext and the secret keys, which the size of an encrypted file and secret keys grows linearly with the number of attributes involved in the access policy and the user's attribute set in ours.

7.2 Computation Costs

We make a theoretical analysis about the time of encryption and decryption in Table 4. The performance is analyzed under three aspects which are the computational costs in terms of multiplication, exponentiation and pairing in $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$. Compared with [7, 30] and [17] in Table 4, the proposed scheme takes less time in user.decryption phase than others, because most bilinear pairing calculation is transferred to CP in ours. The existence of three hash functions $\mathcal{O}(\mathcal{H})$ in [1] results in our scheme being more efficient in user.decryption phase.

In addition, the computation costs of the encryption operation and user.decryption operation are presented in Figure 5(c) and Figure 5(d). From Figure 5(c), it can be known that our scheme has some performance disadvantages, such as encryption time. However, this is a compromise to achieve fully policy-hiding. The time of user's decryption in our scheme is greatly short than others due to the outsource operation. Our scheme reduces the overhead on the user side online computation and has a clear



(a)ciphertext-size (b)secret key-size (c)encryption-time (d)user.decryption-time

Figure 5: Evaluation of algorithms

advantage over ABE scheme without outsourcing. However, it can be found that comparing between a scheme that outsources most of its heavy computational operations and other that doesn't is not fair enough. So the comparison of the decryption cost is performed between our scheme and [25] in Figure 5(d).

8 Conclusion

In this paper, a scheme with fully hiding access policy is studied in the cloud storage system. The policy is hidden by converting the access policy and attribute set into vectors, and the scheme is constructed based on IPE technique. In addition, this decentralized MA-ABE scheme with strong resistance to attacks mentioned in [23] and [29] from potential malicious users. Moreover, the decryption is partially outsourced to the third party proxy services which results in a more efficient decentralizing MA-ABE. Then, the security of the presented scheme is reduced to the DBDH assumption and the DLIN assumption instead of others strong assumptions. We also confirm the scalability and flexibility of the proposed scheme by numerical experiments. However, our scheme only achieves selectively security. It is left as the future work to construct the decentralizing MA-ABE with adaptive security and full hiding policy.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China under Grant (61872087), the National Cryptography Development Fund under Grant (MMJJ20180209), the National Key Research and Development Program of China under Grants No. 2017YFB0802002, International S&T Cooperation Program of Shaanxi Province No. 2019KW-056, the Plan For Scientific Innovation Talent of Henan Province under Grant 184100510012, and the Program for Science and Technology Innovation Talents in the Universities of Henan Province under Grant 18HASTIT022.

References

[1] S. Belguith, N. Kaaniche, and M. Laurent, "Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot," *Computer Networks*, vol. 133, pp. 141–156, 2018.

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

[3] M. Chase, "Multi-authority attribute based encryption," *Proceedings of Theory Cryptography Conference (TCC'07)*, vol. 4392, pp. 515–534, 2007.

[4] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," *The 16th ACM Conference on Computer and Communications Security*, vol. 14, pp. 121–130, 2009.

[5] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of 14th ACM Conference Security in Computing and Communications*, pp. 456–465, 2007.

[6] 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, 2014.

[7] Y. Fan, X. Wu, and J. Wang, "Multi-authority attribute-based encryption access control scheme with hidden policy and constant length ciphertext for cloud storage," in *IEEE Second International Conference on Data Science in Cyberspace*, pp. 205–212, 2017.

[8] A. Ge, J. Zhang, R. Zhang, C. Ma, and Z. Zhang, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2319–2321, 2013.

[9] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proceedings of the 20th USENIX Conference on Security*, pp. 523–538, 2011.

[10] J. Han, W. Susilo, Y. Mu, and J. Yan, "Security analysis of a privacy-preserving decentralized key- policy attribute-based encryption scheme," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2150–2162, 2012.

[11] S. Honhenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography (PKC'14)*, pp. 293–310, 2014.

[12] C. Jin, X. Feng, and Q. Shen, "Fully secure hidden ciphertext policy attribute-based encryption with short ciphertext size," in *ACM International Conference on Advanced Computing, Networking and Security*, vol. 11, no. 6, pp. 91–98, 2016.

[13] J. Lai, R. H. Deng, C. Guan, and J. Wang, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, pp. 1343–1354, 2013.

[14] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *ACM Symposium on Information Computer and Communications Security*, pp. 18–19, 2012.

[15] A. Lewko, and B. Waters, "Decentralizing attribute-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 6632, pp. 568–588, 2011.

[16] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of attribute-based access control with user revocation in cloud data storage," *International Journal of Network Security*, vol. 18, no. 5, pp. 900–916, 2016.

[17] Y. Michalevsky and M. Joye, "Decentralized policy-hiding abe with receiver privacy," *European Symposium on Research in Computer Security Springer*, vol. 11099, pp. 548–567, 2018.

[18] S. Muller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *International Conference on Information Security and Cryptology*, vol. 5461, pp. 20–36, 2009.

[19] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *International Conference on Applied Cryptography and Network Security*, vol. 5037, pp. 111–129, 2008.

[20] T. Okamoto and K. Takashima, "Adaptively attribute-hiding (hierarchical) inner product encryption," *Advances in Cryptology*, vol. 7237, pp. 591–608, 2012.

[21] T. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute based encryption under standard assumptions," *IEEE Transactions on Information Forensics Security*, vol. 11, pp. 35–45, 2016.

[22] H. Qian, J. Li, and Y. Zhang, "Privacy-preserving decentralized ciphertext-policy attribute-based encryption with fully hidden access structure," in *International Conference on Information and Communications Security*, vol. 8233, pp. 363–372, 2013.

[23] Y. Rahulamathavan, S. Veluru, J. Han, F. Li, M. Rajarajan, and R. Lu, "User collusion avoidance scheme for privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Computers*, vol. 65, no. 9, pp. 2939–2946, 2016.

[24] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Advances in Cryptology*, vol. 3494, pp. 457–473, 2005.

[25] J. Shao, Y. Zhu, and Q. Ji, "Efficient decentralized attribute-based encryption with outsourced computation for mobile cloud computing," *IEEE International Symposium on Parallel and Distributed Processing with Applications and IEEE International Conference on Ubiquitous Computing and Communications*, vol. 1, pp. 417–422, 2017.

[26] M. Wang, Z. Zhang, and C. Chen, "Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 4, pp. 1237–1245, 2016.

[27] R. Xu and B. Lang, "A cp-abe scheme with hidden policy and its application in cloud computing," *International Journal of Cloud Computing*, vol. 4, pp. 279–298, 2015.

[28] Z. Ying, J. Ma, and J. Cui, "Partially policy hidden CP-ABE supporting dynamic policy updating," *Journal on Communications*, vol. 36, pp. 178–189, 2015.

[29] L. Zhang, P. Liang, and Y. Mu, "Improving privacy-preserving and security for decentralized key-policy attributed-based encryption," in *IEEE Access*, pp. 1–1, 2018.

[30] H. Zhong, W. Zhu, Y. Xu, and J. Cui, "Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage," in *Soft Computing*, pp. 243–251, 2016.

[31] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 126–138, 2015.

Biography

Leyou Zhang received the M.S. and Ph.D. degrees from Xidian University, in 2002 and 2009, respectively. He is currently a Professor with Xidian University. His current research interests include cryptography, network security, cloud security, and computer security.

Juan Ren received the B.S. degree in mathematics from Nanchang Hangkong University, China, in 2017. He is currently pursuing the Ph.D. degree in applied mathematics with Xidian University, China. His current interests include applied cryptography and cloud security.

Li Kang is a master degree student in the school of mathematics and statistics, Xidian University. Her research interests focus on computer and network security.

Baocang Wang received the B.S. degree in Computational Mathematics and the Application Software, the M.S and the Ph.D. degrees in cryptography from Xidian University in 2001, 2004, and 2006, respectively. He is currently a professor with the School of Telecommunications Engineering, Xidian University. His main research interests include public key cryptography, wireless network security, and cloud computing security.

Fast Scalar Multiplication Algorithms Based on 5P+Q of Elliptic Curve over $GF(3^m)$

Shuang-Gen Liu, Xiang Wang, Yao-Wei Liu, and Dong-Juan Li
(Corresponding author: Shuang-Gen Liu)

School of Cyberspace Security, Xi'an University of Posts and Telecommunications
Xi'an 710121, China
Email: liusgxupt@163.com

(Received Jan. 13, 2020; Revised and Accepted Sept. 8, 2020; First Online Apr. 19, 2021)

Abstract

To improve the efficiency of the very time-consuming operation of scalar multiplication in elliptic curve cryptography (ECC), lots of fast and secure algorithms have been proposed. This paper presents an improved algorithm for calculating 5P directly by using the idea of trading multiplication for squares in terms of jacobian coordinates, which reduces the costs of computing 5P from 15M+10S to 8M+16S. Then, an algorithm for calculating 5P+Q is proposed, which combines the idea of Co_Z point addition. In addition, an addition chain called Improved Fibonacci Type Addition Chain (IFTAC) by taking advantage of the properties of the Generalized Fibonacci sequence is proposed. When the key length is 160 bits, the corresponding addition chain has a chain length of 65, which significantly reduces the length of the chain than the previous algorithm. Then, a new scalar multiplication algorithm based on the properties of IFTAC for elliptic curves is proposed, which protects against Simple Power Attack (SPA) since it iterates 5P+Q in every circulation.

Keywords: Elliptic Curves; Ellptic Curve Cryptography; Generalized Fibonacci Sequence; Scalar Multiplication; Simple Power Attack

1 Introduction

Elliptic Curve Cryptography (ECC) was introduced by Neal Koblitz [12] and Victor Miller [20] independently, it has been widely used in resource-limited hardware environments for its short key size requirement and efficient arithmetic. It is also widely adopted to encryption, decryption, digital signature and verification [8,9]. Its safety is based on the difficulty of elliptic curve discrete logarithm problem (ECDLP), and there are only two faster methods to solve this problem: Baby-step-Gaint-step and Pollard's ρ. The Security of RSA is based on the Integer Factorization Problem (IFP), but there are many efficient algorithms to solve IPF, such as ordinary number field screening method. Therefore, the security of ECC algorithm is much higher than RSA algorithm especially uses same key length. For example, the 160-bit ECC key can provide the same security as the 1024-bit RSA key [23]. The most time-consuming operation in ECC is scalar multiplication (kP), where k is a great integer and P is a point elliptic curve. As the core operation of ECC, the speed of scalar multiplication directly determines the efficiency of ECC implementation [10, 19].

The optimization of scalar multiplication can usually be studied from two aspects:

1) The scalar multiplication operation on the bottom layer such as field inverse, multiplication and so on is optimized to reduce the computational complexity of the underlying filed;

2) Because of scalar k is a greate positive integer generally so we can trading the calculation kP for some basic point operations on the elliptic curve by effectively decompose the scalar k.

Besides, this is an effective way to improve the efficiency of the bottom layer operation that used coordinate transformation to avoid complex inversion operation. In general, by converting the point operations from affine coordinates to projective coordinates, the field inverse with a great computation can be converted into field multiplication with smaller amount of calculation, thus effectively reducing the computation of scalarmmultiplication. Jacobian coordinates system is a commonly used projective coordinates.

There are a lot of algorithms using Jacobian coordinate system to improve the efficiency of scalar multiplication operation for elliptic curve on bottom layer. Dimitrov *et al.* [3] proposed an algorithm for directly calculating 2^kP and 3^kP. Joye [11] improve the point addition and double operation by using the idea of trading multiplication for squares. Mishra [21] firstly uses division polynomials to propose algorithms for directly calculating 5P and 5^kP, which improves the computational efficiency of the

underlying field. Meloni [19] proposed the point addition formula in Jacobian coordinates by using the idea of Co_Z. Longa [18] improved the calculation efficiency of 3P, 5P, 7P, 2P+Q, 3P+Q and other operations by using Meloni's point addition formula and the idea of transform multiplication into square. Lai *et al.* [14] gave an improved algorithm for calculating 7P, 7^kP, 5P and 5^kP with Jacobian coordinate system in prime field GF(P). All of these algorithms reduce the computational complexity of the bottom layer as well as accelerate the efficiency of scalar multiplication of elliptic curves.

The idea of Co_Z is first proposed by Meloni [19], which improves the computational efficiency of point addition, doubling and scalar multiplication on elliptic curves. In recent years, goundar, Wei *et al.* [6, 26] use the skill to optimized scalar multiplication formula on Weierstrass elliptic curve. Lin and Zhang [15] use this skill to improve the computational efficiency of multi-scalar multiplication. Wei *et al.* [27] used Co_Z technique firstlly to improved the point addition and point doubling formula of Montgomery algorithm for elliptic curve over finite field $GF(3^m)$.

For the top layer, there are lots of algorithms for calculating the elliptic curve scalar multiplication kP that can resists Simple Power Attack (SPA) attacks effectively. SPA is a type of side channel attack proposed by literature [13] that derives information about the private key k by analyzing the power consumption trajectory in the encryption process. Since the calculation formula of the point addition and double point operation in the elliptic curve is different, the power consumption trajectory generated during the calculation is also different, so it is easy for attackers to analyze the operation performed in the calculation from the power trajectory to get the information of private key k. SPA is popular among attackers because of its low cost, easy execution and obvious effect. Two methods are usually used to resist SPA attacks, one is to use indistinguishable addition and doubling operations in scalar multiplication algorithm such as addition chain [4, 7], the other is to regularize scalar multiplication algorithm, such as double-and-add algorithm [25] and Montgomery ladder algorithm [2].

Our contributions in this paper are divided three levels:

- We presents an improved algorithm for calculating 5P directly by using the idea of trading multiplication for squares over prime field in terms of Jacobian coordinates. Additionally, combining with the idea of same z-coordinate, an algorithm for calculating 5P+Q directly is proposed.
- Based on the properties of generalized Fibonacci sequence, a new addition chain called Improved Fibonacci Type Addition Chain (IFTAC) is presented which has shorter chain length.
- A new scalar multiplication algorithm by using the properties of IFTAC is proposed, which can resist SPA naturally. The new scalar multiplication algorithm is 22.81% faster than GRAC-258 algorithm in the best case.

2 Background

2.1 Elliptic Curve Cryptography

An elliptic curve E over a field K is defined by the general Weierstrass equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

where $a_1, a_2, a_3, a_4, a_6 \in F$ and $\Delta \neq 0$,theΔ is discriminant of E. Over prime fields of large characteristic, the elliptic curve in the Weierstra form can be simplified to:

$$y^2 = x^3 + ax^2 + b \tag{2}$$

where $a, b \in K, a \neq 0$ and $\Delta = -a^3b \neq 0$.

Now convert points in affine coordinates to corresponding points in Jacobian projective coordinates. In jacobian coordinates, the point P = (X,Y,Z) on E correspond to the affine point $P' = (x, y) = (\frac{X}{Z^2}, \frac{Y}{Z^3})$ where $Z \neq 0$. Then, the curve E in jacobian coordinates is given by:

$$Y^2 = X^3 + aXZ^4 + bZ^6 \tag{3}$$

In the coordinates, the field operation of elliptic curve E involves four basic operations: addition, subtraction, multiplication and square. Let S and M to represent the square and multiplication operation on the finite field respectively. In general, $S/M = 0.6 \sim 0.8$, let's assume that $S = 0.8M$ as well as addition and subtraction operations in the $GF(3^m)$ are neglected. The formulae of point addition and doubling is shown as follows:

Addtion: $P = (X_1, Y_1, Z_1) \in E, Q = (X_2, Y_2, Z_2) \in E, P \neq \pm Q$, Then $P + Q = (X_3, Y_3, Z_3)$ can be evaluated as:

$$\begin{cases} X_3 = u^2 - v^3 - 2X_1Z_2^2v^2 \\ Y_3 = u(X_1Z_2^2v^2 - X_3) - Y_1Z_2^3v^3 \\ Z_3 = vZ_1Z_2 \end{cases} \tag{4}$$

where $u = Y_2Z_1^3 - Y_1Z_2^3$ and $v = X_2Z_1^3 - X_1Z_2^3$. Then the cost of addition is 12M+4S.

Addition with same Z-coordinate: In [19], Meloni considers the case of adding two (different) points having the same Z-coordinate. When points P and Q share the same Z-coordinate, say $P = (X_1, Y_1, Z)$ and $Q = (X_2, Y_2, Z)$, then $P + Q = (X_3, Y_3, Z_3)$ is computed by:

$$\begin{cases} X_3 = D - B - C \\ Y_3 = (Y_2 - Y_1)(B - X_3) - Y_1(C - B) \\ Z_3 = Z(X_2 - X_1) \end{cases} \tag{5}$$

where $A = (X_2 - X_1)^2$, $B = X_1A$, $C = X_2A$, $D = (Y_2 - Y_1)^2$. Obviously, the cost of Co_Z addition involves 5M+2S.

Doubling: $P = (X_1, Y_1, Z_1) \in E, P \neq -P$, then $2P = (X_3, Y_3, Z_3)$ is given by:

$$\begin{cases} X_2 = (3X_1 + aZ_1^4)^2 - 8X_1Y_1^2 \\ Y_2 = (3X_1 + aZ_1^4)(4X_1Y_1^2 - X_2) - 8Y_1^4 \\ Z_2 = 2Y_1Z_1 \end{cases} \quad (6)$$

Then the cost of doubling is 4M+6S.

2.2 Scaled Projective Coordinates System

It is known, [24], that every ordinary elliptic curve over $GF(3^m)$ with a point of order three can be written in the form [5]:

$$E_b : y^2 = x^3 + x^2 + b \quad (7)$$

for some $b \in GF(3^m)$.

The ordinary elliptic curve E_b where $b \neq 0$, via the map $(x, y) \mapsto (u, v)$ defined by:

$$\mathrm{x} = \mathrm{d} * (\mathrm{u} + \mathrm{v}), \mathrm{y} = \mathrm{d} * (\mathrm{u} - \mathrm{v}) \quad (8)$$

is birationally equivalent to Hessian curve $H_d : u^3 + v^3 + 1 = duv$, where $d^3 = -\frac{1}{b}$.

In the projective model, the point $P = (X, Y, Z)$ is correspond to the affine point $P' = (x, y) = (\frac{X}{Z}, \frac{Y}{Z})$, where $Z \neq 0$. Now, we suggest to use the scaled projective coordinate system (X, Y, T), where $T = \frac{Z}{d}$ and (X, Y, Z) is a projective point on E_b.

Let $E_b : Y^2Z = X^3 + X^2Z + bZ^3$, where $b = -\frac{1}{d^3}$. Then the addition and doubling can be shown as follows:

Addtion: Let $(X_3, Y_3, T_3) = (X_1, Y_1, T_1) + (X_2, Y_2, T_2)$, which is the addition in the scaled projective coordinates system. Then, the doubling can be evaluated as:

$$\begin{cases} A_1 = X_1 + Y_1, B_1 = X_1 - Y_1, A_2 = X_2 + Y_2 \\ B_2 = X_2 - Y_2, D = B_1 * T_2, E = A_2 * T_1 \\ F = A_1 * T_2, G = B_2 * T, H = D * E \\ I = F * G, J = F * I, K = E * H \\ X_3 = D * H + J - G * I - K \\ Y_3 = X_3 + J + K \\ T_3 = (\frac{1}{d}) * (D + F - E - G)^3 \end{cases} \quad (9)$$

The cost of addition algorithm is 10M+1C+D, where D is the cost of a field multiplication by the constant $\frac{1}{d}$. When $T_1 = 1$, the cost of addition reduced by 8M + 1C + 1D.

Doubling: Let $(X_3, Y_3, T_3) = [2](X_1, Y_1, T_1)$, which is the doubling in the scaled projective coordinates system. Then, the doubling is given by:

$$\begin{cases} A = X_1 + Y_1, B = X_1 - Y_1 \\ D = (T_1 - A)^3, E = (B - T_1)^3 \\ F = B * D, G = A * E \\ H = T_1 * (D + E) \\ X_3 = F + G \\ Y_3 = F - G \\ T_3 = H \end{cases} \quad (10)$$

The cost of doubling algorithm is 3M + 2C.

2.3 Euclid Addition Chain

Definition 1. *An addition chain computing an integer k is given by a sequence $v = (v_1, v_2, ..., v_s)$ where $v_1 = 1$,$v_s = k$ and $\forall 1 \leq i \leq s$ (s is the length of the addition chain). $v_i = v_{i_1} + v_{i_2}$ for some i_1 and i_2 lower than i [19].*

Definition 2. *An Euclid addition chain computing an integer k is an addition chain satisfies $v_1 = 1$,$v_2 = 2$,$v_3 = v_1 + v_2$ and $\forall 3 \leq i \leq s - 1$, if $v_i = v_{i-1} + v_j$ for some $j < i-1$, then $v_{i+1} = v_i + v_{i-1}$ (called big step) or $v_{i+1} = v_i + v_j$ (called small step) [19].*

3 Improved Algorithm for Calculation of 5P Directly

For the first time by Mishra and Dimitrov [21], it is proposed that calculate 5P directly under the Jacobian coordinates by using the idea of the division polynomial is as mentioned below. Let P=(X : Y : Z) be a point on the elliptic curve. Let Q=5P have coordinates (X_5, Y_5, Z_5). Then X_5, Y_5 and Z_5 can be computed as follows:

$$\begin{cases} X_5 = X_1V^2 - 2Y_1UW \\ Y_5 = Y_1[E^3(12VL^2 - V^2 - 16L^4) - 64TL^5] \\ Z_5 = Z_1V \end{cases} \quad (11)$$

where T = $8Y_1^4$, $M = 3X_1^2 + aZ_1^4$, $E = 12X_1Y_1^2 - M^2$, $L = ME - T$, $U = 4Y_1L$, $V = 4TL - E^3$, $N = V - 4L^2$, $W = EN$. The costs of computing 5P by using this algorithm are 15M+10S.

Improve the above algorithm by using equation $2ab = (a + b)^2 - a^2 - b^2$, which can trading one multiplication for three squares. Therefore, we can use this equation appropriately to reduce the amount of calculation for calculating 5P directly. Let $Z'_5 = 2Z_5 = 2Z_1V$, $X'_5 = 4X_5$, $Y'_5 = 8Y_5$ be convenient for derivation, then $(X'_5, Y'_5, Z'_5) \sim (X_5, Y_5, Z_5)$. Let $E = 6[(X_1 + {Y_1}^2)^2 - X_1^2 - {Y_1}^4] - M^2$, $L = (M + E)^2 - M^2 - E^2 - 2T$, $U = 2TL - E^3$, $V = U - L^2$, $I = U^2 - V^2 + L^4$. By substituting these equations, we can get a new algorithm for calculating 5P directly.

Algorithm 1 An algorithm for calculating 5P

Input: $P = (X_1, Y_1, Z_1)$
Output: $Q = (X'_5, Y'_5, Z'_5)$
1: T $\leftarrow 8Y_1^4$; $M \leftarrow 3X_1^2 + aZ_1^4$;
2: $E \leftarrow 6[(X_1 + {Y_1}^2)^2 - X_1^2 - {Y_1}^4] - M^2$
3: $L \leftarrow (M + E)^2 - M^2 - E^2 - 2T$
4: $U \leftarrow 2TL - E^3$; $V \leftarrow U - L^2$
5: $I \leftarrow U^2 - V^2 + L^4$
6: $W \leftarrow 4[(Y_1^2 + L)^2 - {Y_1}^4 - L^2][(E + V)^2 - E^2 - V^2]$
7: $X'_5 \leftarrow 4X_1U^2 - W$
8: $Y'_5 \leftarrow 4Y_1[E^3(6I - U^2 - 2L^4) - UL^4]$
9: $Z'_5 \leftarrow (Z_1 + U)^2 - {Z_1}^2 - U^2$
10: **return:** $Q = (X'_5, Y'_5, Z'_5) \sim (X_5, Y_5, Z_5)$

Six multiplications operation can be trading for six square operation as well as two multiplication can be reduced though Algorithm 1. The computational complexity of each step is analyzed in Table 1.

Table 1: Operational complexity of Algorithm 1

Step	Computation	Known Terms
T	2S	–
M	M+3S	–
E	2S	X_1^2, Y_1^2
L	2S	M^2
U	2M	E^2
V	S	–
I	3S	–
W	M+2S	L^2, Y_1^4, V^2
X'_5	M	U^2, W
Y'_5	3M	E^3, L^4, I
Z'_5	S	Z_1^2

In Algorithm 1, the costs of computing 5P requires 8M+16S. Compared with the algorithm computation is 15M+10S in Mishra [23], it reduces 7M and increases 6S, the efficiency of the new algorithm is improved by 9.57%, which is also 3.70% and 0.95% faster than Longa [18] and Lai [14] respectively.

4 Algorithm for Calculating 5P+Q

Let the set $K^3\backslash\{(0:0:0)\}$ be a set of non-zero three-dimensional vectors on the finite field $GF(3^m)$, The projection points on the set have the following relation:

$$(X:Y:Z)=\{(\lambda^c X:\lambda^d Y:\lambda Z):\lambda\in K^*\} \tag{12}$$

Therefore, we can make the two points $P=(X_1,Y_1,Z_1)$ and $Q=(X_2,Y_2,Z_2)$ on the elliptic curve have the same Z coordinates by coordinate operations.

The two points P and Q on the elliptic curve in the Jacobian projective coordinates are expressed in affine coordinates as: $P=(\frac{X_1}{{Z_1}^2},\frac{Y_1}{{Z_1}^3})$, $Q=(\frac{X_2}{{Z_2}^2},\frac{Y_2}{{Z_2}^3})$. After the denominator performs the generalization, the two coordinates can be expressed as:

$$\begin{aligned} P&=(\tfrac{X_1{Z_2}^2}{{Z_1}^2{Z_2}^2},\tfrac{Y_1{Z_3}^3}{{Z_1}^3{Z_2}^3})\\ Q&=(\tfrac{X_2{Z_1}^2}{{Z_2}^2{Z_1}^2},\tfrac{Y_2{Z_1}^3}{{Z_2}^3{Z_1}^3}) \end{aligned} \tag{13}$$

Convert to projective coordinates as follows:

$$\begin{aligned} P&=(X_1{Z_2}^2,Y_1{Z_3}^3,Z_1Z_2)\\ Q&=(X_2{Z_1}^2,Y_2{Z_1}^3,Z_1Z_2) \end{aligned} \tag{14}$$

Then P and Q have the same Z coordinate. Therefore, combined with Algorithm 1 and using the idea of Co_Z, we can get the algorithm of calculating 5P+Q directly as Algorithm 2.

Algorithm 2 An algorithm for calculating 5P+Q

Input: $P=(X_1,Y_1,Z_1)$, $Q=(X_2,Y_2,Z_2)$,
Output: $5P+Q=(X_3,Y_3,Z_3)$

1: Calculate the coordinates of $5P=(X_5,Y_5,Z_5)$ according to Algorithm 1.
2: ${X'}_1\leftarrow X_5{Z_2}^2$, ${Y'}_1\leftarrow Y_5{Z_2}^3$, $Z'\leftarrow Z_5Z_2$
3: ${X'}_2\leftarrow X_2{Z_5}^2$, ${Y'}_2\leftarrow Y_2{Z_5}^3$, $Z'\leftarrow Z_5Z_2$
4: $A\leftarrow {X_2}'-{X_1}'$
5: $B\leftarrow {Y_2}'-{Y_1}'$
6: ${X'}_3\leftarrow B^2-X_1A^2-X_2A^2$
7: ${Y'}_3\leftarrow B(X_1A^2-X_3)-Y_1(X_2A^2-X_1A^2)$
8: ${Z'}_3\leftarrow Z'A$
9: **Return:**$(X'_3,Y'_3,Z'_3)\sim(X_3,Y_3,Z_3)$

The cost of computing 5P+Q requires 20M+20S according to Algorithm 2. Compared with algorithms in Longa [18], the efficiency of the new algorithm can be increased by 13.46%, 12.20%, 2.17% respectively.

5 Scalar Multiplication Algorithm

5.1 Fibonacci Type Sequence

Definition 3. *The generalized Fibonacci sequence is a sequence that satisfies the following equation where $pq\neq 0$ [28]:*

$$\begin{cases} f_0=x, f_1=y\\ f_n=pf_{n-1}+qf_{n-2}\ (n\geq 2) \end{cases} \tag{15}$$

The general term formula for Fibonacci sequence is $f_n=\frac{\alpha^n-\beta^n}{\alpha-\beta}$ where $\alpha,\beta=\frac{p\pm\sqrt{p^2-4q}}{2}$. According to the Definition 3, let p=5, q=1, x=1, y=5, then we get a new sequence belongs of Fibonacci type sequence, ie $\{1,5,26,135,701,3640,18901,...\}$. Therefore, the new Fibonacci type sequence can be defined as follows.

Definition 4. *A new Fibonacci type sequence is a sequence that satisfies $F_1=1$, $F_2=5$ and $F_n=5F_{n-1}+F_{n-2}$. Then, the general term formula for this sequence is given by:*

$$f_n=\frac{\alpha^n-\beta^n}{\alpha-\beta} \tag{16}$$

where $\alpha=\frac{5-\sqrt{29}}{2}$, $\beta=\frac{5+\sqrt{29}}{2}$.

The ratio of the front and back terms of the sequence can be evaluated as:

$$\begin{aligned} \lim_{n\to\infty}\frac{F_n}{F_{n+1}}&=\lim_{n\to\infty}\frac{\alpha^n-\beta^n}{\alpha^{n+1}-\beta^{n+1}}\\ &=\frac{2}{5+\sqrt{29}}\\ &\approx 0.19258 \end{aligned} \tag{17}$$

We assume that $\lim_{n\to\infty}\frac{F_n}{F_{n+1}}=\frac{1}{\sigma}$, where $\sigma=\frac{5+\sqrt{29}}{2}$.

5.2 Improved Fibonacci Type Addation Chain

The private key k required for encryption in ECC is a large positive integer. For example, if the length of k is 160 bits, the corresponding decimal number range is $k \in (2^{159}, 2^{160}-1)$, thus we will not calculate kP directly. The common method is to decompose the positive integer k and design corresponding algorithm. Each cycle of the algorithm only iterates addition, doubling or multiple point operation. The algorithm of this paper mainly uses the properties of the new Fibonacci type sequence proposed above, decomposing k into a sequence, and then calculates the value of kP according to the relationship between the front and back terms in the sequence. The properties used to decompose k are as following:

$$\begin{cases} F_{i+1} = F_{i-1} - 5F_i \\ F_{i+1} = F_i * \frac{1}{\sigma} \end{cases} \tag{18}$$

The designed algorithm called Improved Fibonacci Type Addition Chain (IFTAC) by using the above equation is shown as Algorithm 3. This algorithm is used to decompose a large positive integer into an additive chain. According to the properties of the decomposition process, we will get two sets that is m and f after the execution of Algorithm 3. The set m is the main chain and f set is the auxiliary chain. Therefore, when the sets m and f are known, we can restore the original integer k based on these two sets as well. Since we mainly rely on the set m for recovery, the length of m is the length of addition chain.

Algorithm 3 IFTAC decomposition algorithm for integer k

Input: Positive integer k
Output: $m = [\,]$, $f = [\,]$

$\sigma \leftarrow 0.193$
$F_0 \leftarrow k$
$F_1 \leftarrow F_0 * \frac{1}{\sigma}$
$m \leftarrow m \cup \{1\}$
$i \leftarrow 0$
while true **do**
  $F'_{i+2} \leftarrow F_i - 5 * F_{i+1}$
  if $5 * F'_{i+2} \le F_{i+1}$ and $6 * F'_{i+2} \ge F_{i+1}$ **then**
    $F_{i+1} \leftarrow F'_{i+2}$
    $m \leftarrow m \cup \{1\}$
  else
    $f \leftarrow f \cup \{F'_{i+2}\}$
    $m \leftarrow m \cup \{0\}$
    $F_{i+2} = F'_{i+2} * \frac{1}{\sigma}$
  end if
  if $F_{i+2} \ge 0$ and $F_{i+2} \le 5$ **then**
    $tmp\_1 \leftarrow F_{i+1} - 5 * F_{i+2}$
    $f \leftarrow f \cup \{tmp\_1\}$
    $f \leftarrow f \cup \{F_{i+2}\}$
    $m \leftarrow m \cup \{0\}$
    break;
  end if
  $m \leftarrow m << 1$
end while
return: $m = [m_1, m_2, m_3, ...]$, $f = [f_1, f_2, f_3, ...]$

5.3 New Scalar Multiplication Algorithm

The addition chain corresponding to a large integer k has two chains, of m and f, so we can restore the value of k if the two chains are known naturally. Therefore, when calculating scalar multiplication kP in ECC, we can design an algorithm based on the idea as Algorithm 4 described.

So we now take an example to illustrate the decomposition of a large integer. Let k=9968, then the addition chain corresponding to k shown in Example 1.

Example 1. *Decomposition Process of k=9968*
$F_0 = k = 9968$
$F_1 = \lfloor F_0 * 0.193 \rfloor = 1923$, $m \leftarrow m \cup \{1\}$
$F_2 = F_0 - 5 * F_1 = 353$, $m \leftarrow m \cup \{1\}$
$F'_3 = F_1 - 5 * F_2 = 158$, $5 * F'_3 > F_2$
$F_3 = \lfloor F_2 * 0.193 \rfloor = 68$, $m \leftarrow m \cup \{0\}$, $f \leftarrow f \cup \{158\}$
$F_4 = F_2 - 5 * F_3 = 13$, $m \leftarrow m \cup \{1\}$
$F'_5 = F_3 - 5 * F_4 = 3$, $5 * F'_5 > F_4$
$F_5 = \lfloor F_4 * 0.193 \rfloor = 2$, $m \leftarrow m \cup \{0\}$, $f \leftarrow f \cup \{3\}$
$0 < F_5 < 5$
$tmp\_1 = F_4 - 5 * F_5 = 3$,
$f \leftarrow f \cup \{3\}$,
$f \leftarrow f \cup \{2\}$,
$m \leftarrow m \cup \{0\}$,
$m \leftarrow m << 1$,

Algorithm 4 Scalar Multiplication Algorithm Based on IFTAC

Input: $m = [m_1, m_2, m_3, ...]$, $f = [f_1, f_2, f_3, ...]$
Output: kP

$lm \leftarrow Length(m)$
$lf \leftarrow Length(f)$
$T_{lm} \leftarrow f[lf-1]P$
$k \leftarrow 2$
for $i = lm - 1$ to 0 **do**
  if $m_i = 0$ **then**
    $T_i \leftarrow 5 * T_{i+1} + f[lf-k]P$
    $k \leftarrow k + 1$
  end if
  if $m_i = 1$ **then**
    $T_i \leftarrow 5 * T_{i+1} + T_{i+2}$
  end if
end for
Return $kP = T_0$

Output: $m = [1, 0, 1, 0, 0]$, $f = [158, 3, 3, 2]$

As shown in Example 1, we obtained the the output of algorithm are set $m = \{1, 0, 1, 0, 0\}$ and $f = \{158, 3, 3, 2\}$,and the addition chain of k=9968 is $\{2, 13, 68, 353, 1923, 9968\}$. Any positive integer k will get 2 sets through the algorithm 3. The set m is used to mark the GAP generated in decomposition process. If $m_i = 1$, it means that the ratio of the front and back terms is not significantly offset from $\frac{1}{\sigma}$, that is, GAP is not generated. If $m_i = 0$, it means that GAP is generated in this loop. In this case, we need to adjust the ratio to $\frac{1}{\sigma}$. The set f is mainly used to store the offset when GAP is generated so that the original positive integer k can be recovered eventually and the last element in f stores the last element of the addition chain.

According to the results set m and set f calculated from Example 1, the process of calculating kP=9968P is shown in Example 2.

Example 2. *Calculate 9968P.*

Input: $m = [1, 0, 1, 0, 0]$, $f = [158, 3, 3, 2]$
$lm = 5$, $lf = 4$, $k = 2$
$T_5 = f[3]P = 2P$
$m_4 = 0$, $T_4 = 5 * T_5 + f[2]P = 13P$
$m_3 = 0$, $T_3 = 5 * T_4 + f[1]P = 68P$
$m_2 = 1$, $T_2 = 5 * T_3 + T_4 = 353P$
$m_1 = 0$, $T_1 = 5 * T_2 + f[0]P = 1923P$
$m_0 = 1$, $T_0 = 5 * T_1 + T_2 = 9968P$

Output: $kP = T_0 = 9968P$

6 Analysis of Algorithm

6.1 Length of IFTACs

We randomly select 10000 positive integers k with 160, 192, 224 and 256 bits to evaluate the chain length of IFTAC. When k is 160 bits, the range of generating random numbers is $(0, 2^{160} - 1)$; For k is 192 bits, the range is $(0, 2^{192} - 1)$; For k is 224 bits, the range is $(0, 2^{224} - 1)$; for k is 256 bits, the range is $(0, 2^{256} - 1)$. The statistical algorithm of chain length is recorded in Algorithm 5. Python programming is used to implement algorithms of decomposition and recovery as well as statistics of chain length.

Algorithm 5 Chain length statistical algorithm

1: $l \leftarrow length \in \{160, 192, 224, 256\}$
2: **for** $i = 0$ to 9999 **do**
3: $k \leftarrow RandomInteger(0, 2^l - 1)$
4: $lm \leftarrow lm \cup \{IFTAC(k)\}$
5: $times \leftarrow count(lm)$
6: **end for**
7: Return $times$

Table 2: Chain length statistics of 160bits integers

Algorithm	Chain length
Fibonacci-and-Add [19]	358
Window Fib-and-Add [19]	292
EAC-270 [19]	270
GRAC-258 [7]	258
Fibonacci [29]	230
Pell [29]	125
IFTAC	65

The statistical results of the chain length of 10000 positive random integers are represented by a line graph, as shown in Figure 1. As can be seen from Figure 1, the optimal chain length of k with 160bits is 65, while that of 192 bits, 224 bits and 256 bits positive integer is 79, 92 and 105, respectively. When k is 160 bits, the chain lengths of different addition chain algorithms are shown in Table 2.

From Table 2, the chain length of IFTAC with k of 160bits is shortened by 48% and 71.74% compared to Pell and Fibonacci, which is 74.81% and 75.93% shorter than GRAC-258 and EAC-270, compared to Window Fib-and-Add and Fibonacci-and-Add shortened by 77.74% and 81.84% as well.

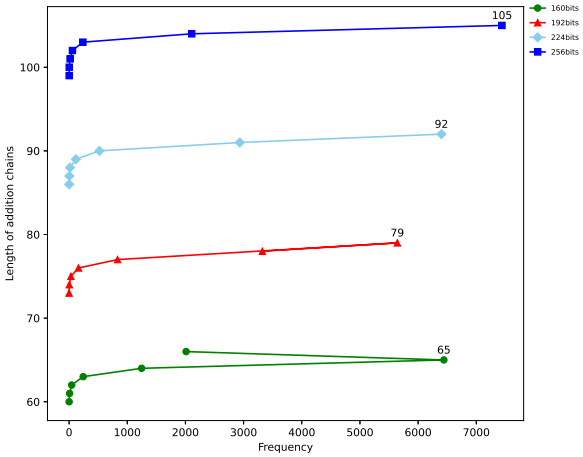


Figure 1: Chain length statistics

6.2 Performance Analysis of Algorithms

In algorithm 5, a 5P+Q operation is iterated once in each cycle, resulting in a fuzzy energy trajectory, so that it can resist SPA attacks. The costs of addition is 8M+1C+1D in the best case and the cost of doubling is 3M+2C. Although the exact proportions between S and M may have vary greatly depending on the field implementation, for field sizes used in cryptography it is quite common to as-

sume $1[S] \approx 0.8[M]$ [1]. Since the characteristic of the elliptic curve is three, so we can simplify cubic computation with Frobenius self-homomorphism. The computing speed of cubing is at least ten times faster than that of multiplication or squaring. [17]. So we will assume that $1[C] \approx 0.1[M]$. And the addition and subtraction can be ignored as well as a field multiplication by a constant, denote by D. Obviously, a 5P+Q operation needs two times doubling and two times addition and the costs of one times doubling and one times additionin scaled projective coordinates are 11M+3C+1D. In Algorithm 4, a 5P+Q operation is required for each round of cycles. We assume that calculating kP requires n cycles, then the costs of n times 5P+Q are 2n*(11M+3C+1D), and 2P needs to be calculated for the first time in the best case. Therefore, When k is an integer of 160 bits, the chain length of IFTAC is 65 and the scalar multiplication is called IFTAC-160, then the cost of kp is 1472M, which is 38.41% and 22.81%faster than OST [16] and GRAC-258 [22] respectively. The costs of each algorithm when k is 160 bits is shown in Table 3.

Table 3: Calculation costs of different algorithms

Algorithm	#[m]
OST [16]	2390
Fibonacci-and-add [19]	2311
Window Fib-add-add [19]	1960
GRAC-258 [22]	1907
IFTAC-160	1472

7 Conclusions

In this paper, we derived an improved algorithm in section 3 for calculating 5P directly ,which costs 8M+16S and improves the efficiency by 9.57%, 6.82%, 0.95% than [18] and [14]. Then, the Algorithm 2 is presented for calculating 5P+Q directly by using the idea of Co_Z, which improved by 13.46%, 12.20%, 2.17% than [18]. In section 4, based on the properties of Fibonacci type sequence, we propose Algorithm 3 to decompose a large positive integer and get the corresponding addition chain. Then, a new fast and secure scalar multiplication algorithm for elliptic curves is presented. The Algorithm 4 can resist SPA since it iterates 5P+Q in each iteration. In section 5, we theoretically analyze the performance of Algorithm 3 and Algorithm 4. When k is an integer of 160 bits, the chain length of IFATC is 65, which is shortened by 48%, 71.74%, 74.81% and 75.93% compared to Pell, Fibonacci, GRAC-258 and EAC-270 as well as 77.74% and 81.84% shortened by Window Fib-and-Add and Fibonacci-and-Add. The cost of kp is 1472M by using scaled projective coordinates, which is 38.41% and 22.81% faster than OST [16] and GRAC-258 [22] respectively. The idea of Co_Z and coordinate transformation can be widely used to improve the efficiency of scalar multiplication of elliptic curve. In the later research, we will focus on finding a coordinate system or elliptic curve that is more suitable for Co_Z operation to improve the efficiency of the bottom layer.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61872058), the Key Research and Development Program of Shaanxi (Program No.2021NY-211). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] R. M. Avanzi, "Aspects of hyperelliptic curves over large prime fields in software implementations," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 148–162, 2004.

[2] P. N. Chung, C. Costello, and B. Smith, "Fast, uniform, and compact scalar multiplication for elliptic curves and genus 2 jacobians with applications to signature schemes," *arXiv preprint arXiv:1510.03174*, 2015. (`https://eprint.iacr.org/2015/983.pdf`)

[3] V. Dimitrov, L. Imbert, and P. K. Mishra, "Efficient and secure elliptic curve point multiplication using double-base chains," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 59–78, 2005.

[4] Y. Dosso, F. Herbaut, N. Méloni, and P. Véron, "Euclidean addition chains scalar multiplication on curves with efficient endomorphism," *Journal of Cryptographic Engineering*, vol. 8, no. 4, pp. 351–367, 2018.

[5] R. R. Farashahi, H. Wu, and C. A. Zhao, "Efficient arithmetic on elliptic curves over fields of characteristic three," in *International Conference on Selected Areas in Cryptography*, pp. 135–148, 2012.

[6] R. R. Goundar, M. Joye, and A. Miyaji, "Co-Z addition formulæ and binary ladders on elliptic curves," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 65–79, 2010.

[7] R. R. Goundar, K. I. Shiota, and M. Toyonaga, "Spa resistant scalar multiplication using golden ratio addition chain method," *International Journal of Applied Mathematics*, vol. 38, no. 2, pp. 83–88, 2008.

[8] L. Han, Q. Xie, and W. Liu, "An improved biometric based authentication scheme with user anonymity using elliptic curve cryptosystem.," *International Journal Network Security*, vol. 19, no. 3, pp. 469–478, 2017.

[9] G. Hou and Z. Wang, "A robust and efficient remote authentication scheme from elliptic curve cryptosystem.," *International Journal Network Security*, vol. 19, no. 6, pp. 904–911, 2017.

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

[11] M. Joye, "Fast point multiplication on elliptic curves without precomputation," in *International Workshop on the Arithmetic of Finite Fields*, pp. 36–46, 2008.

[12] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[13] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*, pp. 388–397, 1999.

[14] Z. X. Lai and Z. J. Zhang, "Research on algorithm of underlying layer of elliptic curve in jacobian coordinate system," *Bulletin of Science and Technology*, vol. 31, no. 10, pp. 244–248, 2015.

[15] Q. Lin and F. Zhang, "Efficient precomputation schemes of kP+lQ," *Information Processing Letters*, vol. 112, no. 11, pp. 462–466, 2012.

[16] H. Liu, Q. Dong, and Y. Li, "Efficient ECC scalar multiplication algorithm based on symmetric ternary in wireless sensor networks," in *Progress in Electromagnetics Research Symposium-Fall (PIERS-FALL'17)*, pp. 879–885, 2017.

[17] S. G. Liu, R. R. Wang, Y. Q. Li, and C. L. Zhai, "An improved ternary montgomery ladder algorithm on elliptic curves over gf (3ˆ m)," *International Journal Network Security*, vol. 21, no. 3, pp. 384–391, 2019.

[18] P. Longa, "Accelerating the scalar multiplication on elliptic curve cryptosystems over prime fields," *Computer Science*, US20090074178A1, 2007.

[19] N. Meloni, "New point addition formulae for ecc applications," in *International Workshop on the Arithmetic of Finite Fields*, pp. 189–201, 2007.

[20] V. S. Miller, "Use of elliptic curves in cryptography," in *Conference on the theory and application of cryptographic techniques*, pp. 417–426, 1985.

[21] P. K. Mishra and V. Dimitrov, "Efficient quintuple formulas for elliptic curves and efficient scalar multiplication using multibase number representation," in *International Conference on Information Security*, pp. 390–406, 2007.

[22] S. C. Pang, S. F. Liu, Z. F. Cong, and L. Z. Yao, "An efficient scalar multiplication algorithm on montgomery-form elliptic curve," *Acta Electronica Sinica*, vol. 39, no. 4, pp. 865–868, 2011.

[23] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[24] N. P. Smart and E. J. Westwood, "Point multiplication on ordinary elliptic curves over fields of characteristic three," *Applicable Algebra in Engineering, Communication and Computing*, vol. 13, no. 6, pp. 485–497, 2003.

[25] P. Srinate and B. Chiewthanakul, "A variant of the schnorr signature using an elliptic curve over a field of characteristic two," in *The 15th International Joint Conference on Computer Science and Software Engineering (JCSSE'18)*, pp. 1–5, 2018.

[26] J. Wei, X. Liu, H. Liu, and W. Guo, "A low-time-complexity and secure dual-field scalar multiplication based on co-Z protected NAF," *IEICE Electronics Express*, vol. 11, no. 11, pp. 20140361–20140361, 2014.

[27] W. Yu, B. Li, K. P. Wang, W. X. Li, and S. Tian, "co-Z montgomery alogorithm on elliptic curves over finite fields of characterrstic three," *Chinese Journal of Computers*, vol. 40, no. 5, pp. 1121–1133, 2017.

[28] K. Zhang, X. Xu, and W. Y. Li, "Properties and generalization of generalized fibonacci sequences," *Science and Technology Innovation Herald*, no. 16, pp. 230–230, 2013.

[29] N. Zhang and S. Tan, "Elliptic curve scalar multiplication based on fibonacci number," in *The 5th International Conference on Intelligent Networking and Collaborative Systems*, pp. 507–510, 2013.

Biography

Shuang-Gen Liu received the PH.D. degree in cryptography form Xidian University in 2008.He is currently an associate professor with the school of cyber security, Xi'an University of Posts and Telecommunications, Xi'an,China. He is a member of the China Computer Federation, and a member of the Chinese Association for Cryptologic Research.His recent research interests include crptography and information security.

Xiang Wang received the B.Eng. Degree in communication engineering from Xi'an University of Posts and Telecommunications, ShanXi, China, in 2018, where she is currently working towards the M.Eng. Degree in information security. Her research interests include elliptic curve cryptosystem and scalar multiplication algorithms.

Yao-Wei Liu has participated in the ssctf network attack and defense competition held by seclover security company and participated in the first training of its network security college. At present, he is studying in the information security major of Xi'an University of Posts and telecommunications. His research interests are network attack and defense and web security.

Dong-Juan Li is now an undergraduate at Xi'an University of Posts and Telecommunications, China. Her main research direction is information security. Such as, cryptography, network security and so on are being studied.

Large-Scale Social Network Privacy Protection Method for Protecting K-Core

Jian Li[1], Xiaolin Zhang[1], Jiao Liu[1], Lu Gao[1], Huanxiang Zhang[2], and Yueyang Feng[3]
(Corresponding author: Xiaolin Zhang)
School of Information Engineering, Inner Mongolia University of Science and Technology[1]
Baotou 014010, China
School of Computer Engineering and Science, Shanghai University[2]
Shanghai 200000, China
School of Science, Inner Mongolia University of Technology[3]
Hohhot 010010 , China
Email: fyylijian@outlook.com
(Received Feb. 3, 2020; Revised and Accepted Sept. 7, 2020; First Online May 30, 2021)

Abstract

Social network analysis has many important applications and methods which depend on the sharing and publishing of graphs. For example, link privacy requires limiting the probability of an adversary identifying a target sensitive link between two individuals in the published social network graph. However, the existing link privacy protection methods have low processing power for large-scale graph data and less consideration of community protection in the publishing graphs. Therefore, aiming at sensitive link privacy protection, a large-scale social network privacy protection model to protect K-Core (PPMPK) was proposed. The large-scale social network graph was processed to ensure that the core number and the community structure of the nodes were unchanged based on the Pregel parallel graph processing model. Extensive experiments on the real data sets showed that the proposed method could effectively process the large-scale graph data and protect the data availability of the published graphs, especially in community protection.

Keywords: Community Protection; K-Core; Pregel; Privacy Protection; Social Network

1 Introduction

With the development of the social network, the Internet plays an increasingly important role in daily life. Online Social Network saves a large number of users' personal information. For example,the facebook's daily active users reached 1.401 billion in 2017 and generated PB-level data every hour. These information contains both sensitive and nonsensitive data. The large-scale social network graph data are saved in the cloud environment which provide a trusted mobile terminal and a secure environment [7, 8] before user privacy sensitive data are anonymity. How to protect the privacy of sensitive data has attracted more and more researchers' attention. The researchers Mining the information in the social network can discover the hidden structure of the network. Community structure is an important feature and widespread in the real-world social network.

Typically, a community is a collection of nodes in the network that have the same or similar roles [9, 12]. However, people who use social network applications are always facing serious privacy breaches and malicious attacks. Each element that makes up a social network may lead to the disclosure of user privacy information. The main privacy of social network includes node attribute privacy and edge privacy. At present, the study on social network privacy protection methods has made positive progress. Researchers have proposed various privacy protection methods for different social network privacy issues. Node attribute privacy includes node identification or sensitive attributes such as name, phone number, address, etc. Researchers used data generalization [20], perturbation [13], or adding noise nodes [15] to protect node attribute privacy.

For edge privacy, researchers proposed different privacy protection models to protect the edge between nodes according to different attack backgrounds. Random perturbation technique, which randomly modifies the original data to reduce the attacker's inferred confidence is able to resist complex background attacks. In this paper, we decompose social network graph to get the node core number. The information interchange between nodes based on Pregel model. The edges are simultaneously perturbed according to the probability while ensured that the core number of nodes are unchanged in the social network graph. Our main contributions can be summarized as follows:

1) Aiming at dealing with large-scale social network graphs, we propose a search result of reachable nodes (SRRN) algorithm that quickly finds the neighbor nodes' information based on the Pregel model.

2) According to the neighbor information, we propose a large-scale social network privacy protection model (PPMPK) for protection of K-Core. The K-Core decomposition graph is used to measure the influence of the node. The edges is perturbed in the social network while protecting the core number of the nodes.

3) Rigorous experiments have been performed on five real world social network data sets. The promising experimental result has demonstrated that the proposed approach can achieve comparably good results when compared with the previous privacy protection approaches. These results also verify the efficacy of protecting the community structure.

The rest of the paper is organized as follows. Section 2 reviews the related works on protecting social networks' privacy and Section 3 details the define preliminaries. Section 4 proposes the privacy protection model. The specific algorithm steps are presented in Section 5. The experimental evaluation is performed in Section 6. Section 7 concludes the paper and points out the future work.

2 Relate Works

At present, most of the edge privacy protection methods adopt anonymous model to prevent the leakage of private information and malicious attacks. It is an efficient anonymous model to protect sensitive edge based on random perturbation technology. In Reference [16], the random probability model was used to implement privacy protection by adding, deleting and switching edges. This method caused the perturbed graph to be a random graph or even a useless graph. In order to improve the efficiency of anonymous graphs, Zhang [19] proposed the perturbation based on the subgraph structure. This method divided the original network graph into several subgraphs and randomly perturbed the edge in subgraphs which increased the degree of some nodes. The probability that these nodes are identified in the subgraph was increased. To solve this problem, a formula that balances the degree of nodes was applied to reduce the probability [5]. Ford [6] proposed a local neighbor random perturbation algorithm running on a single workstation. This method randomly selected the local neighbors of the source node to replace the target node to protect the sensitive edges. In Reference [20], a distributed random perturbation algorithm was designed to solve the problem of large-scale social networks. Furthermore, a sensitive area random perturbation algorithm was provided to improve data availability [17].

Community structure is an important feature of social network graph. Anonymous social networks while protecting the community structure has become a hot topic. The spectrum of the graph serves as an important topological feature of the graph. Ying and Wu [14] compared the similarity between nodes, randomly adding and deleting k edges. This method protected the spectrum of graph changing to protect the community structure. Campan [4] used the community partition algorithm based on graph segmentation theory to added and deleted edges by calculating the Laplacian matrix. Zheng [21] used differential privacy for privacy protection to the online social network structure query, and protected the social network community structure while anonymization. Kumar [9] used the upper approximation concept of rough sets to divide the community for anonymity, but the algorithm execution efficiency is low. Zhang [18] provided a method to anonymited the dynamic social network while protected the community structure as possible.

In summary, the random perturbation algorithm can protect the edge privacy well. Anonymous methods for protecting community structure based on community division have a problem of low processing efficiency for large-scale social network graph. Therefore, a new solution is proposed. After random perturbance anonymity, the influence of the node is maintained, then the stability of the community is guaranteed. The k-core decomposition graph is used to measure the influence of the node. Seidman [12] first proposed the concept of k-core and gave the k-core decomposition process. Kitsak [1] studied the relationship between the k-core and the speed of disease propagation in the social network. Experiments showed that the k-core has more reliable results than the node degree and betweenness centrality. In Reference [11], a privacy protection algorithm maintained the core number of nodes unchanged by the operation of adding, deleting, and switching edges on single workstation. To this end, the proposed privacy protection model PPMPK differs from the traditional numerical perturbations or graph modification methods in that it paralleling selects neighbor nodes for perturbation. At last, the result protected the community's stability under the conditions of privacy.

3 Related Definitions

Let G = (V, E) denotes a directed social graph, consisting of a set of vertices V and a set of edges E. $V = \{v_1, v_2, \cdots, v_n\}$, each of these nodes corresponds to a real user in the social network. $E_{ij} =< v_i, v_j >$ indicates a directed social edge by user i to user j, where i is the source node and j is the destination node.

Definition 1. *(R-neighbor) r is a given non-negative integer. node u, v are two different node in the graph. Dist(u, v) represents the shortest path length of node u to node v. if node v satisfies the condition, node v is said to be the r-neighbor of node u:*

$$Dist(u, v) \leq r \tag{1}$$

Definition 2. *(K-Core) Let k be an integer. The degree of any node v of the set C∈V is not less than k, and*

the largest subgraph G_c (C, E |C) derived therefrom is called K-Core. The largest subgraph that obtained after recursively removing the nodes with degrees less than k and the edges expected to be connected is the K-Core.

Definition 3. *(Core number) If node v belongs to the k-core and does not belong to the $(k+1)$-core, the core number of node v is k. The k-core decomposition is to decompose the social network graph into the largest subgraph from the border of the social network to the central. Each node in the social network graph will have a core number after decomposition.*

Definition 4. *(K-shell) A group of nodes with the core number of k is called a k-shell.*

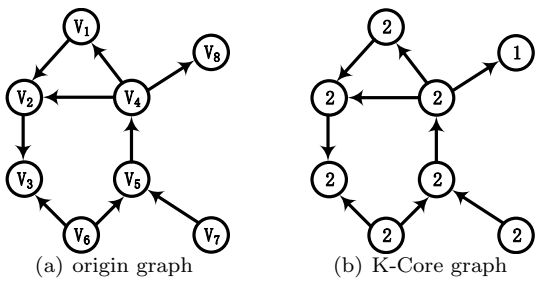
Definition 5. *(Ef_degree) In the k-core decomposition process, the number of the high core or the same code nodes connected is called node effective degree.*

Definition 6. *(K-lamina) In the K-Core decomposition process, nodes with higher effective degree than k in the same k-shell are called k-lamina type nodes.*

Definition 7. *(K-corona) In the k-core decomposition process, nodes with the equal number of effective degree and core number in the same k-shell are called k-corona type node.*

4 Large-Scale Social Network Privacy Protection Model for Protecting K-Core

In the social network, if you want to hide the existed edge <u, v>, you only need to hide the source node u or the destination node v. Only the source or destination node is known, the existence of the edge cannot be inferred [6]. Therefore, a large-scale social network privacy protection model PPMPK for protecting k-cores is proposed to anonymous social network. The main idea is to preserve the edge $< u, v >$ with a probability $p(0 \le p \le 1)$. If the probability of the edge <u, v> is $(1-p)$, the algorithm replaces each of the node with node w. The privacy level of the published graph can be adjusted according to the probability p. In order to get a privacy protection model, some examples and properties are given first.



(a) origin graph (b) K-Core graph

Figure 1: Origin graph and K-Core decomposition graph

Example 1. *Figure 1(a) is the original social network. Figure 1(b) is k-core decomposition graph with node label which is the core number of nodes. As shown in Figure 1(a), it is assumed that the edge $<v_4, v_8>$ is deleted. v_4 is 2-corona node and v_8 is 1-corona type node. As defined by the k-core, the decomposition graph first decomposes the shell of the lower core number nodes. Node v_8 and connected edges are first decomposed. The deleted edges have no effect on node v_4 because v_4 is a higher core number compared to the node v_8. As defined by k-corona, the core number of k-corona nodes is equal to the number of ef_degree, so the core number of node v_8 is downgraded.*

Property 1. *The edge that a higher core number node connected a lower core number node is deleted. The K-Core of the higher core number node is not affected. In lower core number nodes, the core number of k-corona type nodes is downgraded, but the k-lamina type nodes is not downgraded.*

Example 2. *As shown in Figure 1(a), it is assumed that the edge$<v_2$,$v_3>$ is deleted. v_2 is 2-corona and v_3 is 2-lamina. As defined by k-lamina and k-corona, the ef_degree of k-lamina type node is higher than core number. if the edge $<v_2$,$v_3>$ is deleted, the core number of node v_2 will unchange. The core number of v_3 will downgraded.*

Property 2. *The edge that a higher core number node connected a lower core number node is deleted. The k-core of the higher core number node is not affected. In lower core number nodes, the core number of k-corona type node is downgraded, but the k-lamina type node is not downgraded.*

Example 3. *As shown in Figure 1(a), it is assumed that the edge $<v_6, v_3>$ is deleted. v_6 and v_3 are k-corona type nodes with same core number. According to the k-corona definition, the ef_degree of the node is equal to the core number. If the edge $<v_6, v_3>$ is deleted, the core number of nodes v_6 and v_3 are both downgraded.*

Property 3. *The edge of k-corona type node connection with the same-core is deleted. The core number of nodes are both downgraded.*

Lemma 1. *(K-Core perturbance safety condition) For each edge, it is divided into 3 conditions by node type. If only $\forall <u,v> \in E$ satisfies the following conditions while perturbing the edge by choosing candidate node w, the core number of nodes does not change.*

1) $\forall$core[u]>core[v], $\forall$w$\in$V:core[w]$\geq$core[v]$\wedge$(w,v)$\notin$E;
2) $\forall$core[u]=core[v]$\wedge$ef_degree[u]>core[u],$\forall$w$\in$V: core[w]$\geq$core[v]$\wedge$(w,v)$\notin$E;
3) $\forall$core[u]=core[v]$\wedge$ef_degree[u]=core[u]$\wedge$ef_degree[v] =core[v],$\forall$w,q$\in$V:core[w]$\geq$core[v]$\wedge$(w,v)$\notin$E, core[q]$\geq$core[u]$\wedge$(q,u)$\notin$E;

Definition 8. *In the anonymous process of the directed graph G={ V, E}, the K-Core perturbance safety conditions (KPSC) is the sufficient condition for implementing the random perturbation model for protecting the K-Core.*

The KPSC condition 1, (∀core[u]>core[v]) indicates that the edge connection type is the higher core number node connected to the lower core number node. (∀w∈V: core[w]≥core[v]∧(w, v)∉E) shows the candidate node w which is the neighbor of node u and not connected to node v, then edge <u, v> is deleted and <w, v> is added. For high core number nodes u and w, deleting and adding edges that connected lower core number has no effect on the nodes. For the lower core number node v, deleting and adding a high-core number edge, the core number of nodes has no effect.

The KPSC Condition 2,(∀core[u]=core[v]∧ef_degree[u] >core[u]) indicates that the edge type is connected to the same core node. there exists a k-lamina type node. (∀w∈V: core[w]≥core[v]∧(w,v)∉E) expresses the candidate node w which is the higher core number neighbor of node u and is not connected to node v.

- If the *k*-lamina type nodes connected to the k-corona type node, the neighbor of k-corona type node is selected as a candidate. The KPSC condition 3 proves that the core number of perturbed k-corona type nodes has no effect.

- If the *k*-lamina type nodes connected to the k-lamina type nodes. the total number of edges are unchanged in the perturbation. In the same k-shell, the k-lamina type nodes with a smaller number of connected edges are preferentially decomposed. That is, the k-lamina type nodes with added edges will be decomposed, so adding edges has no effect on the core number of nodes.

The KPSC condition 3, (∀core[u]=core[v]∧ef_degree[u] =core[u]∧ef_degree[v]=core[v]) indicates that the edge is connected to the same-core k-corona type node. (∀w, q∈V: core[w]≥core[v]∧(w, v)∉E, core[q]≥core[u]∧ (q, u)∉E) indicates the condition for the selection of candidate nodes. For the k-corona type node with the same core number, ef_degree[v]=core[v] means that if an edge is deleted, the core number will downgraded, so the perturbance is twice. The nodes connected to the same core number are in the same shell. The k-core decomposition first decomposes the k-corona type node and the edge connected to it, so deleting and adding the connection edge has no effect on the k-corona type candidate node.

Based on the above, the KPSC condition can protect the core number of nodes while perturbing the three type edges in the social network graph, so the goal of protecting the k-core must satisfy the KPSC in the process of the perturbance. The KPSC is implemented based on a sufficient condition for protecting the large-scale social network privacy protection model of the k-core. A large-scale social network privacy protection model that protects the k-core for a given directed graph G=$\{V, E\}$, the perturbation neighbor r, the probability p. The main steps are:

1) The algorithm decomposes the social network graph to get the node core number and finds the r reachable neighbor of the node.

2) The probability is assigned to the edge in the graph. If the assignment probability is p, the edge is preserved. If the assignment probability is 1-p, the edge is perturbed. The perturbance edge is determined according to the edge retention probability. The candidate nodes are selected according to the node core number. The candidate nodes selection process satisfies the KPSC.

5 Large-Scale Social Network Privacy Protection Algorithm for Protecting K-Core

In order to obtain the candidate nodes that meet the random perturbation of directed graph. A node reachability search K-Core algorithm is proposed. The algorithm performs a reachable list. Each node generates a random neighbor table (RNT). As shown in Table 1, the RNT consists of (srcid, dstid, hops, core, ef_degree) and each row of the RNT list is an Random Neighborhood Table Entry (RNTE), Table 1 is the initialization RNT list in Figure 1 (such as 11022 indicates that the node v_1 is initialized with a core number of 2 and the effective degree of 2). In the RNTE:

- Source node (srcid): Source node id;
- Destination node (disid): When the node is initialized, the label value is the source node id. During the information transmission, the label value is the source node reachable destination node id.
- Hops: The number of hops indicates that the node needs several iterations to pass information to the destination node;
- Core number: The number of k-core of the nodes;
- Effective degree(ef_degree): The number of effective degrees of the node.

5.1 Node Reachability Search Algorithm based on K-core

The Search results of reachable nodes (SRRN) algorithm is based on the Pregel model. The Pregel model searches for reachable nodes through message transmission, message merging, and message processing functions. At last, SRRN generates a list of nodes RNTs.We present the pseudo-code of SRRN processes in Algorithm 1.

The SRRN algorithm initializes the RNT list of the node to determine whether the node is an active node

Table 1: Initialize the node RNT table

Node	RNTE
V_1	11022
V_2	22023
V_3	33022
V_4	44023
V_5	55023
V_6	66022
V_7	77011
V_8	88011

Algorithm 1 Search Results of Reachable Nodes (SRRN)

Input: Social network graph G, reachable parameter r
Output: Node reachable list

```
1: Initialization: RNT (G);
2: SuperStep=0;
3: while SuperStep ≤ r do
4:   for v_i∈V∩v_i is active do
5:     for Message from vi.srcRNT do
6:       Merge(Message);
7:     end for
8:     if Message.IsNotExist(v_i.Attr) then
9:       Update RNT.hop+1;
10:      Update RNT.srcid=vi.id;
11:    else
12:      VoteToHalt();
13:    end if
14:    if New AddMessage in Message then
15:      Send New AddMessage to vi.disid;
16:    end if
17:  end for
18: end while
```

according to whether the node has out degree. If the node status is inactive, the node don't need to calculate. If the node status is active, the update messages insert according to the messages are not exist in the RNT list and send the updated messages to the destination node.

5.2 Perturbation Algorithm that Keeps the Core Number of Nodes

Algorithm 2 shows the core preserving perturbation of node (CPPN) algorithm. The CPPN algorithm obtains the forward and back RNT list information of the node through the SRRN algorithm. The CPPN algorithm uses the random function to determine whether to perform edge perturbation. If the assignment probability is p, the edge is preserved. If the assignment probability is 1-p, the random perturbation edge of the candidate node is selected. The edge connection condition is determined by the RNTE value of the node. Finally, the random edge perturbation is performed according to different situations.We present the pseudo-code of SRRN processes in Algorithm 2.

Algorithm 2 Perturbation Algorithm for core preserving perturbation of node (CPPN)

Input: Social network graph G, perturbance probability P
Output: Perturbation edge result list

```
1: Initialization: SRRN(G); SRRN(G.reverse);
2: ResultList=∅;
3: for (u,v)∈ E do
4:   if Random==P then
5:     ResultList← (u,v);
6:   else
7:     if core[u]>core[v] or core[u]<core[v] then
8:       High_To_Lower(u,v);
9:     end if
10:    if ef_degree[u]==ef_degree[v]∩ef_degree[u]
       ==core[u] then
11:      Corona_To_Corona(u,v);
12:    else
13:      Lamina_To_Lamina(u,v);
14:    end if
15:  end if
16: end for
17: return ResultList;
```

It is judged whether the edge is perturbed based on the probability function. If the perturbation is required, the edge is perturbed according to the different connecting situation. If the high-core node is connected to the low-core node, and the algorithm 3 is selected. The algorithm 4 is selected when the k-corona node with the same-core is connected. If the K-lamina type node with the same core number is existed in the connected edge , the algorithm 5 is selected. Algorithm 3 is a perturbation edge algorithm for high-core connected low-core. Lines 3-7 and 18-22 ensure that the core number of nodes is unchanged and the reachability between nodes is guaranteed. Lines 8 and 23 indicate that if no candidate nodes can guarantee the reachability between the nodes, then the candidate nodes in the opposite direction are selected to ensure that the core number is unchanged.

Figure 2 shows the results of perturbing the edges that high-core nodes connect to low-core. Table 2 shows the RNT list of nodes (only the RNTE value of the perturbed nodes are listed, the bolds are the back propagation RNTE value, and the RNTE (1) represents the 1-neighbor of node). If perturbing the edge $<v_4, v_8>$, the algorithm compares the core number of nodes v_4 and v_8 in the RNT list. The result indicates that the connected edge is a high-core connected low-core. Deleting the connected edge $<v_4, v_8>$, the algorithm selects the reachable neighbor of the high-core node v_4. The candidate nodes are $\{v_5, v_6, v_7, v_1, v_2, v_3\}$, and the nodes $\{v_1, v_2, v_3\}$ are preferentially selected. The preferentially selected nodes ensure the core number of nodes unchanged while ensuring the reachability between nodes. Finally, the algorithm randomly selects the node v_1, and adds an

Algorithm 3 High-core connection low-core perturbance (High_To_Lower)

Input: An edge <u,v>
Output: The perturbed edge result

```
1: Initialization: candidate=∅; Array=∅;
2: if core[u]>core[v] then
3:    for u.reverse(RNT) do
4:       if core[candidate]≥core[v] and candidate isNo-
         tExist in v.RNT then
5:          Add candidate to Array;
6:       end if
7:    end for
8:    if Array.Size==0 then
9:       for u.RNT do
10:         if core[candidate] ≥core[v] and candidate is-
            NotExist in v.RNT then
11:            Add candidate to Array;
12:         end if
13:      end for
14:   end if
15:   return ResultList←(Random(Array),v);
16: else
17:   for v.RNT do
18:      if core[candidate] ≥core[u] and candidate isNo-
         tExist in v.RNT then
19:         Add candidate to Array;
20:      end if
21:   end for
22:   if Array.Size==0 then
23:      for v.reverse(RNT) do
24:         if core[candidate] ≥ core[u] and candidate is-
            NotExist in v.RNT then
25:            Add candidate to Array;
26:         end if
27:      end for
28:   end if
29:   return ResultList←(u,Random(Array));
30: end if
```

edge$<v_1, v_8>$; For the perturbed edge $<v_7, v_5>$, the algorithm compares the core number of nodes v_7 and v_5 in the RNT list. The RNT list indicates that the connected edge type is a low-core connected high-core. Finally, the algorithm deletes the connected edge $<v_7, v_5>$, and selects the neighbor of the high-core node v_5. The candidate nodes have $\{v_6, v_4, v_1, v_2, v_8\}$. The algorithm randomly selects node v_6, and adds edges $<v_7, v_6>$.

Algorithm 4 is k-corona type nodes perturbation algorithm. Lines 2-8 are the first perturbation, and ensure that the core number of node v does not change. Lines 10-16 are the second perturbation, and ensure that the core number of node u does not decrease.

Figure 3 shows the result of the connected edge perturbation of the k-corona node. Table 3 shows the RNT list of the perturbed nodes, the perturbed edge is $<v_6, v_3>$. The core number and effective degree of nodes v_6 and v_3 are compared in the RNT list. The result shows that

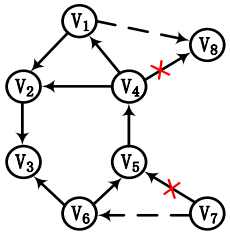


Figure 2: High-core connection low-core perturbance results

Table 2: The RNT list of nodes

Node	RNTE(1)	RNTE(2)
V_4	45122	46222
	41122	47211
	42123	**43222**
	48111	
V_8	**84123**	**85222**
V_5	56122	**52223**
	54123	**51222**
	57111	**58222**
V_7	75122	74223

Algorithm 4 K-Corona nodes perturbation (Corona_To_Corona)

Input: An edge <u,v>
Output: The perturbed edge result

```
1: Initialization: candidate=∅; Array=∅;
2: for u.RNT do
3:    if core[candidate] ≥core[u] and candidate isNotEx-
      ist in v.RNT then
4:       Add candidate to Array;
5:    end if
6: end for
7: ResultList ← (Random(Array),v);
8: candidate=null; Array=null;
9: for v.RNT do
10:   if core[candidate] ≥core[v] and candidate isNotEx-
      ist in v.RNT then
11:      Add RNT.disid to Array;
12:   end if
13: end for
14: ResultList←(u,Random(Array));
15: return ResultList;
```

the edge with same-core and needs twice perturbations. The first perturbation ensures that the core number of nodes v_6 is unchanged. The perturbance selects the high-core neighbors of the node v_3 are $\{v_1, v_2, v_4\}$. The result preferentially selects the node v_2, adds the connection edge $<v_6, v_2>$. The second perturbation guarantees the core number of node v_3 and selects the high-core neighbor v_5, v_4 of node v_6, and prefer to select node v_5. The result adds an edge $<v_5, v_3>$.

Table 3: The RNT list of the perturbed nodes

Node	RNTE(1)	RNTE(2)
V_6	**63122**	**64223**
	65122	
V_3	36122	34223
	32123	31222

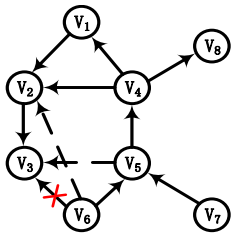


Figure 3: Perturbance result of the same-core K-Corona type node

Algorithm 5 is k-lamina type node perturbation algorithm. The two nodes have at least one k-lamina type. The algorithm first selects the high-core or same-core neighbors of the k-lamina type node. If the neighbor of the k-lamina type node is always selected, the k-lamina type node will become the k-corona type node. At last, the core number of node will downgrade. The anonymous program will update the ef_degree value when run at lines 5-10.

Algorithm 5 Same-core k-lamina node perturbance (Lamina_To_Lamina)

Input: An edge <u,v>
Output: The perturbed edge result ResultList

```
1: if core[u] < ef_degree then
2:    Select u.RNT;
3:    ResultList← (Random(Array),v);
4:    Update(u.ef_degree-1);
5: end if
6: if core[v]< ef_degree then
7:    Select v.RNT;
8:    ResultList← (u,Random(Array));
9:    Update(u.ef_degree-1);
10: end if
11: return ResultList;
```

Figure 4 shows the results of the perturbation of the k-lamina type nodes. Table 4 shows the RNT list of the perturbed nodes. The perturbed edge $<v_4, v_1>$ is required. The core number and the effective degree of nodes v_4 and node v_1 are compared in the RNT list. It shows that nodes v_4 and node v_1 are the k-lamina type node with the same-core. The neighbor of the node v_4 is selected. The candidate nodes have $\{v_3, v_5, v_6\}$. The node v_3 is randomly selected, and the ef_degree of the node v_4 is updated.

Table 4: The results of the perturbation of the k-lamina type nodes

Node	RNTE(1)	RNTE(2)
V_4	45422	46222
	41122	47211
	42123	**43222**
V_1	14123	15222
	12123	**13222**

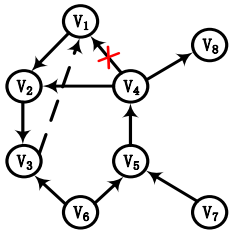


Figure 4: The same core K-lamina type node perturbance results

5.3 Algorithm Security Analysis

PPMPK obtains an anonymous social network graph through random perturbations. G*={V*, E*} is the anonymous result graph obtained by deleting and adding edges under the edge retention probability p of the original social network graph G={V, E}. If the edge assignment probability is 1-p, the edge is deleted. The candidate node is added. Assuming that all edges are perturbed once. Even if there are k-corona-type nodes connected, the two perturbations will increase the security of the connected edges. We suppose the number of nodes and edges do not changed, $|V| = |V^*|$,$|E| = |E^*|$, and $|E|$ is the set of edges in the original social network graph G. If the edge retention probability is 0, the social network graph is transformed into a completely random graph, so that the data availability of the published graph is zero.

In the random perturbation graph, the attacker wants to identify the target node, and must judge the possibility of the connected edge according to the edge retention probability p and the anonymous graph G*. If it is a full graph random perturbance, the candidates for the target node x are all nodes from u∈G* (except v), because adding and deleting edges are completely random. The attacker wants to determine the exact probability of each edge as:

$$P_{r(p)} = \frac{1}{\begin{pmatrix} |E| \\ \frac{|E|}{1-p} \end{pmatrix} \begin{pmatrix} |\overline{E}| \\ \frac{|\overline{E}|}{1-p} \end{pmatrix}} \tag{2}$$

The denominator represents the probability of a random perturbation under the retention probability p. For PPMPK, the candidate node is restricted to the r neighbors. The probability that the attacker can recognize the

edge is proportional to Pr(p) and r value. By increasing the r value and retention probability p, the publisher can effectively prevent the attacker from identifying the real destination node.

6 Performance and Evaluation

The PPMPK privacy protection algorithm performs performance analysis and evaluation. The result compares PPMPK with R-SW and R-A/D algorithms [16], and uses different dimensions and methods for verification.

6.1 Experimental Setup

The algorithms were tested using five real social network datasets:

1) Karate Karate Club Network.
2) Jazz Jazz Musician Network.
3) URV email network.
4) Polblogs political blog data set.
5) Amazon dataset.

The datasets description is shown in Table 5. n is the number of nodes and m is the number of edges. $\overline{deg}$ represents the average degree and d represents the diameter.

Table 5: Data set

Dataset	n	m	$\overline{deg}$	d
Karate	34	78	4.588	5
Jazz	198	2742	27.697	6
URV email	1133	5451	9.622	8
Polblogs	1224	16715	27.312	8
Amazon	403394	2443408	12.114	15

The experimental operating environment: 15 computing nodes, CPU 1.8GHz, 16GB RAM, Hadoop 2.7.2, Spark2.2.0, programming language: Scala 2.11.12.

6.2 Running Time Analysis

Figure 5 is a comparison of PPMPK with R-A/D and R-SW (r = 3) for running time. It can be seen that PPMPK selects 5% to 25% of the edges for anonymity in the 3-hop neighbors of node in the Amazon dataset. When running the anonymous algorithm, the time differences between the R-SW and R-A/D is small. This is because the random algorithm does not need to calculate the candidate nodes, but the PPMPK algorithm needs to decompose the social network to obtaining a K-Core graph and requires information propagation between nodes when selecting candidate nodes, so PPMPK algorithm takes slightly longer than R-SW and R-A/D.

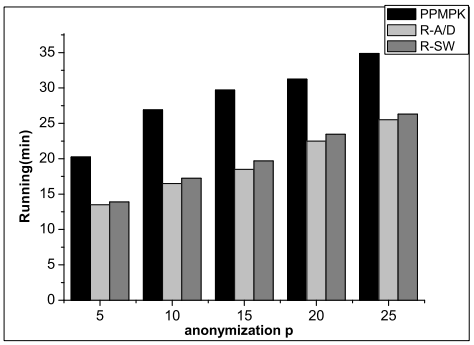


Figure 5: Running time comparison graph

6.3 Information Loss Analysis

We use $RelativeError = |u - u^*|/u$ (u and u^* refer to the metrics in the original graph and the perturbation graph) to evaluate the average shortest path and closeness centrality after the social network graph perturbance. The smaller the value, the smaller the information loss, the better the data availability is maintained.

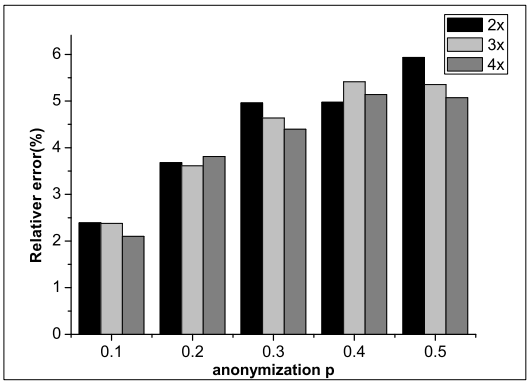


Figure 6: The relative error of average shortest path

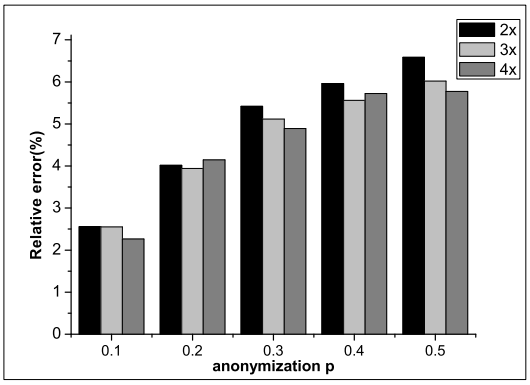


Figure 7: The relative error of closeness centrality

Figures 6 and 7 show different local neighbor r on the average shortest path and closeness centrality in the Polblogs dataset (Note: X-axis represents the percentage of perturbed edges, and Y-axis represents the relative error

rate after the perturbation). As the number of perturbed edges increases, the relative error of the average shortest path and the closeness center are increased. As the perturbation edges increase, the information loss is greater. When the number of perturbed edges is constant, the relative error may decrease as the perturbance r increases because the candidate node is far away from the source node. When adding edges,fewer vertices are changed in the path. In contrast, the distance between the candidate nodes and the source node are greatly change.

In order to measure the influence of the anonymous algorithm on the graph structure, the topological properties of the graph are used to represent the feature changes of the graph, for example: average distance ($\overline{dist}$), diameter (d), transitivity (t), second small eigenvalue (μ^2) of the Laplace matrix, betweenness centrality (C_B) and closeness centrality (C_C). Under the p-perturbance, the information loss of the social network graph is expressed as V(G,G_p^*). V(G) is the topological property value of the original graph, and V(G_p^*) is the topological property of the graph after the perturbance.

$$V(G, G_p^*) = \left\| V(G) - V(G_p^*) \right\|_2 \tag{3}$$

Table 6 shows the information loss of the PPMPK and RA/D and R-SW algorithms after perturbing the edges under different datasets of p=0.25 and r=3 (the bold value is the optimal value). It can be seen from the table that the PPMPK algorithm can protect the structure information of the graph well in most cases. Figure 8 shows the average shortest path change after anonymity in the jazz social network. The closest algorithm to origin graph is the PPMPK algorithm, which indicates that the information loss of the PPMPK anonymity is the smallest.

Table 6: The information loss situation

Methods	$\overline{\textbf{dist}}$	**d**	**t**	$\mathbf{C_B}$	$\mathbf{C_C}$	μ^2
karate						
R-A/D	0.048	0.331	0.031	0.030	0.053	**0.058**
R-SW	0.120	0.596	0.019	0.022	0.037	0.184
PPMPK	**0.020**	**0.120**	**0.011**	**0.019**	**0.026**	0.060
Jazz						
R-A/D	0.188	1.927	0.111	0.008	0.045	1.856
R-SW	**0.127**	0.504	0.105	0.006	**0.032**	**0.056**
PPMPK	0.237	**0.244**	**0.093**	**0.001**	0.102	0.137
URV email						
R-A/D	0.099	0.508	0.038	0.128	0.147	0.317
R-SW	0.114	0.165	0.044	0.088	**0.012**	**0.003**
PPMPK	**0.073**	**0.100**	**0.042**	0.001	0.039	0.047
Ploblogs						
R-A/D	0.116	2.319	0.040	0.144	0.110	0.429
R-SW	0.088	0.846	0.033	0.089	0.038	0.062
PPMPK	**0.010**	**0.600**	**0.001**	**0.001**	**0.022**	**0.058**

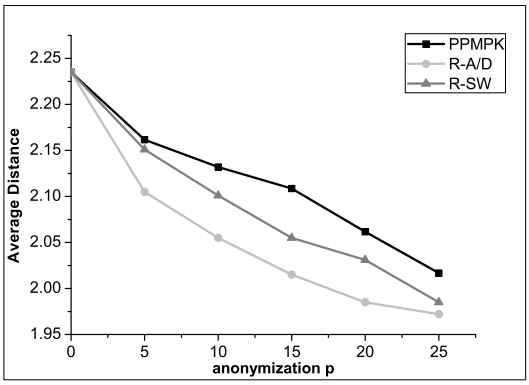


Figure 8: The information loss of average shortest path

6.4 Analysis of Community Structure

By running three kinds of clustering methods to evaluate the influence of the perturbation edge on community, the technology used is python's $\|igraph\|$ package. Community clustering method are Walktrap(WT) [10], Infomap(IM) [2], Multilevel(ML) [3]. Although these algorithms allow overlapping communities, this experiment sets the corresponding parameter to zero. In order to visualize the changes in the community, the precision index (Precision_index) is introduced [11]. If the $l_{tv}(v) = l_{pv}(v)$ is equal to 1. the result indicates that the node in the original graph is the same as the anonymous community. If the node community is not the same as the anonymous community, then $l_{tv}(v) \neq l_{pv}(v)$ is 0. The precision index formula is defined as follows:

$$Precision\_index\,(G, G^*) = \frac{1}{n} \sum\nolimits_{v \in G} \rho_{l_{tv}(v) = l_{pv}(v)} \tag{4}$$

Table 7 shows the community changes of PPMPK, R-A/D and R-SW algorithms after clustering at p=0.25 and r=3. It can be seen from Table 7 that the R-SW algorithm works better than the R-A/D algorithm in protecting the community to which the node belongs, because the R-SW is only switching edge. The PPMPK algorithm considers the node's K-Core for edge perturbation anonymity. The PPMPK algorithm can achieve the best results in most cases. Figure 9 shows the results of the Polblogs dataset using the Infomap clustering algorithm. As the anonymity increases, the accuracy of clustering results decreases. From the overall results, the PPMPK algorithm is more effective than R-SW and R-A/D algorithms in protecting the community.

6.5 Analysis of Large-Scale Data Results

For large datasets, Figures 10 and 11 are the results of the algorithms running in the Amazon dataset. As the anonymity increases, the relative error of graph connectivity and average clustering coefficient increases. This is because the algorithm needs to be perturbed more edges as the anonymity P increases. The R-SW and R-A/D algorithms do not consider the nodes' changes after

Table 7: Perturbed 25% of the edge of community changes

	Method	IM	ML	WT
karate	R-A/D	0.236	0.186	0.327
	R-SW	0.284	0.213	0.323
	PPMPK	**0.114**	**0.117**	**0.147**
Jazz	R-A/D	0.131	0.118	0.113
	R-SW	0.101	**0.099**	**0.078**
	PPMPK	**0.089**	0.142	0.08
URV email	R-A/D	0.313	0.396	0.495
	R-SW	0.251	0.373	0.281
	PPMPK	**0.22**	**0.154**	**0.255**
Polblogs	R-A/D	0.226	0.23	0.227
	R-SW	0.123	**0.062**	0.066
	PPMPK	**0.062**	0.102	**0.023**

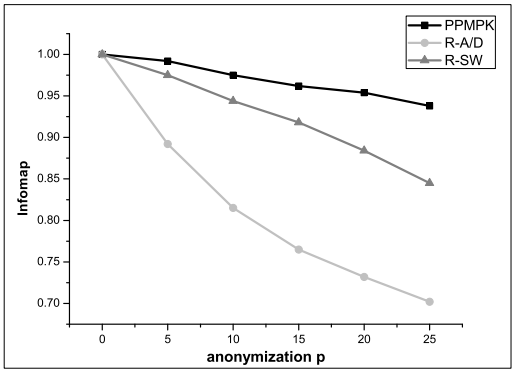


Figure 9: The result of Infomap clustering algorithm

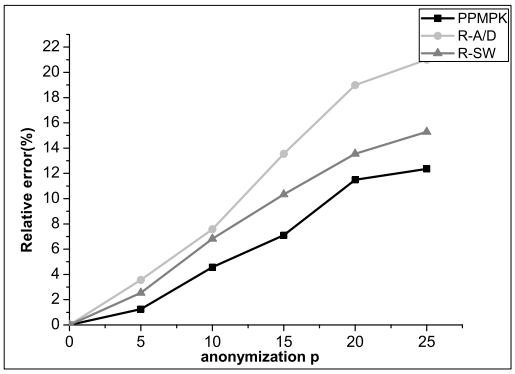


Figure 10: The result of connectivity

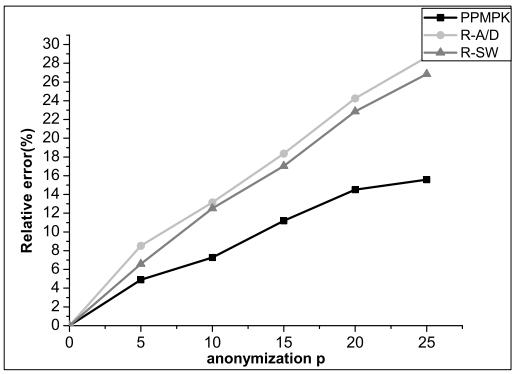


Figure 11: The result of average clustering coefficient

anonymity, so the structural information loss of the graph is larger than the PPMNK algorithm. The information loss of PPMPK algorithm at P=25% is less than 20%, which guarantees the structure of the graph well. It shows that the PPMPK algorithm also has a good performance in large-scale social networks.

7 Conclusion

In order to protect the stability of the node community, a large-scale social network privacy protection method for protecting the k-core is proposed. In order to deal with the large-scale social network graph, the algorithm used Pregel graph calculation model as a basis to compute large-scale social network graphs. The core number was used to measure the influence of the nodes. The core number of nodes were kept constant after anonymity. The community of social network graph were kept stable. Finally, it was verified that the PPMPK algorithm can effectively protect the structural stability of the graph community by experimental tests and analysis while increased the data availability of the graph and ensured the privacy of the social network graph compared with other algorithms.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (No.61562065) and Inner Mongolia Natural Science Foundation (No.2019MS06001). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] R. Assam, M. Hassani, and M. Brysch, "(k, d)-core anonymity: Structural anony-mization of massive networks," in *The 26th International Conference on Scientific and Statistical Database Management*, pp. 1–12, 2014.

[2] V. D. Blondel, J. L. Guillaume, and R. Lambiotte, "Fast unfolding of communities in large networks," *Physics and Society*, vol. 2008, no. 10, pp. 10008, 2008.

[3] B. J. Cai, H. Y. Wang, and H. R. Zheng, "Evaluation repeated random walks in community detection of social networks," in *International Conference on Machine Learning and Cybernetics*, pp. 1849–1854, 2010.

[4] A. Campan, Y. Alufaisan, and T. M. Truta, "Preserving communities in anony-mized social networks," *Transactions on Data Privacy*, vol. 8, no. 1, pp. 55–87, 2015.

[5] A. M. Fard, K. Wang, and P. S. Yu, "Limiting link disclosure in social network anal-ysis through

subgraph-wise perturbation," in *Proceedings of 15th International Conference on Extending Database Technology (ICEDT'12)*, pp. 109–119, 2012.

[6] A. Ford and K. Wang, "Neighborhood randomization for link privacy in social network analysis," *World Wide Web-internet and Web Information Systems*, vol. 18, no. 1, pp. 9–32, 2013.

[7] T. Gao, T. Li, R. Jiang, Y. Ming, and R. Zhu, "Research on cloud service security measurement based on information entropy," *International Journal of Network Security*, nol. 21, no. 6, pp. 1003–1013, 2019.

[8] X. Hui and W. Yang, "Security access solution of cloud services for trusted mobile terminals based on trustzone," *International Journal of Network Security*, vol. 22, no. 2, pp. 201–211, 2020.

[9] S. Kumar and P. Kumar, "Upper approximation based privacy preserving in online social networks," *Expert Systems with Applications*, vol. 88, pp. 276–289, 2017.

[10] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[11] F. Rousseau, J. Casas-Roma, and M. Vazirgiannis, "Community preserving anonymization of graphs," *Knowledge and Information Systems*, vol. 54, no. 2, pp. 1–29, 2018.

[12] S. B. Seidman, "Network structure and minimum degree," *Social Networks*, vol. 5, no. 3, pp. 269–287, 1983.

[13] Y. Sun, Y. Yuan, and G. Wang, "Splitting anonymization: A novel privacy-preserving approach of social network," *Knowledge and Information Systems*, vol. 47, no. 3, pp. 595–623, 2016.

[14] X. Ying and X. Wu, "On link privacy in randomizing social network," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 28–39, 2009.

[15] M. Yuan, L. Chen, and P. S. Yu, "Protecting sensitive labels in social network data anonymization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 633–647, 2013.

[16] X. Ying and X. Wu, "Randomizing social networks: A spectrum preserving approach," in *Proceedings of the SIAM International Conference on Data Mining (SIAM'08)*, pp. 739–750, 2008.

[17] X. L. Zhang, J. Li, and J Liu, "Social network sensitive area perturbance method based on firefly algorithm," *IEEE Access*, vol. 7, pp. 137759–137769, 2019.

[18] X. L. Zhang, J. Liu, and J. Li, "Large-scale dynamic social network directed graph k-in&out-degree anonymity algorithm for protecting community structure," *IEEE Access*, vol. 7, pp. 108371–108383, 2019.

[19] L. Zhang and W. Zhang, "Edge anonymity in social network graphs," in *Proceedings of International Conference on Computational Science and Engineering (ICCSE'09)*, pp. 1–8, 2009.

[20] X. L. Zhang, W. C. Zhang, and C. Zhang, "Dgsperturb: A distributed social privacy protection algorithm based on graph structure perturbation," *Journal of Computers (Taiwan)*, vol. 28, no. 5, pp. 51–61, 2017.

[21] X. Zheng, Z. Cai, and G. Luo, "Privacy-preserved community discovery in online social networks," *Future Generation Computer Systems*, vol. 93, pp. 1002–1009, 2019.

Biography

Jian Li received bachelor's degree in software engineering from Inner Mongolia University of Science and Technology, Inner Mongolia, China, in 2017. At present, he is pursuing a master degree in computer science and technology at Inner Mongolia University of Science and Technology, China. His research areas include big data processing technology, social network privacy protection technology.

Xiaolin Zhang received bachelor's degree in computer science and technology from Northeastern University in 1988, and master's degree from Beijing University of Science and Technology in 1995, and Ph.D. in computer science and technology from Northeastern University in 2006. Since 1988, she has worked in Inner Mongolia University of Science and Technology. her research areas include big data processing, social network privacy protection, XML database, XML data stream, wireless sensor network, uncertain database, flame image database.

Jiao Liu received the BS degree in medical information engineering from Taishan Medical College, Shandong, China, in 2017. Currently, she is pursuing a master's degree in computer science and technology at Inner Mongolia University of Science and Technology. Her research interests include big data processing technology, social network privacy protection technology.

Lu Gao received her master's degree from Inner Mongolia Agricultural University. She is an associate professor at Inner Mongolia University of Science and Technology. Her research interests include large-scale social network privacy protection.

Huanxiang Zhang received her master's degree from Inner Mongolia University of Science. She is a Ph.D. student at Shanghai University. Her research interests include large-scale social network privacy protection and artificial intelligence.

Yueyang Feng received the BS degree in mathematics from Tianjin University of Technology and Education, Tianjin, China, in 2017. At present, she is pursuing a master's degree in mathematics at Inner Mongolia University of Technology, China.

Unsupervised Data Anomaly Detection Based on PCA-oritened Deep Auto-encoder Network

Rui Yang[1,2] and Dong Ye[1]
(Corresponding author: Dong Ye)
School of Electrical Engineering, Zhengzhou University of Science and Technology[1]
Henan Intelligent Information Processing and Control Engineering Technology Research Center[2]
Zhengzhou 450000, China
Email: 910675024@qq.com; 352720214@qq.com
(Received Mar. 22, 2020; Revised and Accepted Oct. 21, 2020; First Online May 30, 2021)

Abstract

Using the traditional auto-encoder (AE) model to detect data anomaly has certain limitations. This is because data must be preprocessed to obtain a data-set containing only normal data. In this paper, the difference between the output and input data of each reconstruction of AE is calculated by combining the principal component analysis method, and the abnormal data is isolated. Namely, it divides the input data into normal and abnormal parts. AE reconstructs the normal part, and the abnormal part is optimized by the proximal method. Then an unsupervised data anomaly detection method based on the depth auto-coding network (DAE) model is implemented using the alternating direction multiplier method to train the whole model and achieve the predetermined training times before the output results. Finally, we conduct experiments on real data set and compare the related state-of-the-art deep learning models with our proposed model. The results show that the AUC value of our proposed model reaches the optimal value. As a result, the performance of data anomaly detection is better, which has a good practical value for the detection of data anomalies in real problems.

Keywords: Auto-encoder; Deep Learning; Unsupervised Data Anomaly Detection; Principal Component Analysis

1 Introduction

With the popularization of the Internet and the development of information technology, anomaly detection has gradually become a research hotspot in the field of data mining. Anomaly detection, which aims to detect abnormal values in observed data, is widely used in credit card fraud, network intrusion detection, equipment anomaly detection, medical analysis and weather forecasting. In the above fields, the generation of abnormal data, relative to the large number of normal data, is usually regarded as a random phenomenon [13,21]. It does not conform to the data pattern of normal data, does not have the data correlation of positive sample. Meanwhile, the abnormal data samples are relatively rare. Therefore, how to accurately detect a small number of abnormal data in a large number of data has become a key difficulty to be solved in this research field [12,17].

In recent years, researchers have done a lot of research work on data anomaly detection. Data anomaly detection methods can be roughly divided into two methods: traditional machine learning and deep neural network [6,9]. Currently, the main machine learning methods for anomaly detection include Isolation Forest (IF) and One Class Support Vector Machine (OCSVM) [1,3]. For example, Susto *et al.* [16] adopted IF data anomaly detection method, and the AUC value of anomaly detection in power dispatching flow data reached to 0.968. Shang *et al.* [15] used the improved OCSVM method to detect network traffic anomalies, which improved the classification accuracy by nearly 10%, and effectively solved the disadvantages of traditional traffic anomaly detection methods, such as low accuracy and high cost. At present, Auto Encoder (AE) and Variational Auto Encoder (VAE) deep neural network models are mainly used in data anomaly detection [5,7,14]. Ji *et al.* [10] used the improved variator auto-encoder to detect network intrusion data set KDD-CUP, and the AUC value reached to 0.951, higher than the traditional data anomaly detection model.

With the progress of deep neural network model for data anomaly detection, the performance of traditional machine learning model is not as good as that of neural network model. For example, due to the classification feature of IF and OCSVM model, it is no longer suitable for high dimensional data anomaly detection. Although auto-encoding and variational auto-encoding models generally perform better than traditional machine learning models in data anomaly detection, model training requires normal (clean) data [2, 8]. On the other hand, the traditional auto-coding model is used to data anomaly detection. The training data set generally includes normal

data and abnormal data. If the model can only be trained with normal data, data preprocessing must be carried out in advance to obtain the data set containing only normal data, which has certain limitations on anomaly detection of actual data.

In the construction of auto-encoding network, the deep learning method has been used to detect abnormal data in a specific environment. Aiming at the problems existing in the current data anomaly detection methods, this paper introduces the auto-encoding model in the Deep learning network and improves the auto-encoding model to build the Deep Auto Encoder (DAE) model. The model not only outperforms the traditional machine learning model in anomaly detection, but also overcomes the limitation that the traditional auto-encoding requires normal data for model training. And the network model can be trained without normal (clean) data. The experimental results show that the deep auto-coding model is better than the traditional machine learning model and the auto-encoding model in data anomaly detection.

2 Construction of Deep Auto-encoder Model

The deep auto-encoder model proposed in this paper is based on the auto-encoder model, and the data is divided into two parts for processing by principal component analysis, which can make up for the defect that the traditional auto-encoder model needs normal data sets for training and detecting abnormal data effectively.

2.1 Auto-encoder Model Network

AE is a three-layer neural network model containing input layer, hidden layer and output layer. It obtains important information in data through unsupervised learning algorithm. The low-dimensional compression expression process from the input layer to the hidden layer can be called the Encoder stage. And the compression feature map in the hidden layer restores approximate raw data of the output layer, this process is called Decoder stage [18].

Let the original spatial data be $R^{m\times n}$, m is the data instance number in the original space. n is the dimension of each instance data. $X^{(i)} \in R^n (i = 1, 2, \cdots, m)$, the encoding and decoding processes are shown in Equation (1) and Equation (2), respectively.

$$h^{(i)} = E_\theta(x^{(i)}) = \sigma(Wx^{(i)} + b). \quad (1)$$

$$\hat{h}^{(i)} = D_\theta(h^{(i)}) = \sigma(W'x^{(i)} + b). \quad (2)$$

Where $h^{(i)}$ gets the feature expression of the hidden layer through the encoder for each input instance $x^{(i)}$. $\theta = (w, b)$ is the network parameter, and W is the weight matrix before the hidden layer. b is the bias value of neurons in the hidden layer. $\sigma(x)$ is the activation function. In this paper, we select the Sigmoid function as Equation (3). In formula (2), $\hat{h}^{(i)}$ is the reconstructed expression of the hidden layer after decoding. $\theta = (w', b')$ is the network parameter. W' is the weight matrix from the hidden layer to the output layer, usually $W' = W^T$. b' is the bias value of the neuron in the output layer.

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (3)$$

The learning goal of AE is to minimize the value of reconstruction error L, so that the input and output values are as close as possible. The choice of error function L is the mean square error loss function shown in Equation (4).

$$L(x, \hat{x}) = \frac{1}{m}\sum_{1}^{m}(\hat{x} - x)^2. \quad (4)$$

Then the target function of auto-encoder can be rewritten as:

$$min_{D,E} = ||X - D(E(X))||. \quad (5)$$

Where X is the input data. E is the encoder. D is the decoder. $||\cdot||$ is L2-norm. The deep auto-encoder model mentioned in this paper is improved on the basis of the original auto-encoder. In fact, it improves Equation (5) to make it better be applied to data anomaly detection.

2.2 Deep Auto-encoder Network Construction

The traditional auto-encoding network is applied to data anomaly detection. Firstly, a auto-encoder is trained with the normal data set, and the error of abnormal data reconstruction is calculated by the auto-encoder [20]. Finally, it sets a threshold α. If the reconstruction error is greater than this threshold, it is abnormal. Otherwise, it is normal. It can be seen that the traditional auto-encoder is applied to data anomaly detection, and normal (clean) data is needed for model training. Moreover, according to the objective function of Equation (5), the objective function has no constraint term (regular term), which also makes the model prone to over-fitting.

Principal component analysis (PCA) mainly finds the eigenvector corresponding to the maximum eigenvalue of the covariance matrix in the data set. Thus it finds several directions with the largest data variance, and achieves the effect of dimensionality reduction for the data. In the process of data anomaly detection, PCA method is introduced and the input data is divided into two parts: normal data and abnormal data. The output of the normal data is reconstructed by the auto-encoding network, and the output of the abnormal data is optimized by the proximal method, as shown in Figure 1.

At the initial moment, assuming that the abnormal part of the input data is 0, that is, the entire initial input data X_0 is treated as a normal part. After the auto-encoding network training, the difference between the output reconstructed data L_{D0} and the initial input data X_0 is the abnormal part S_0 of the current training.

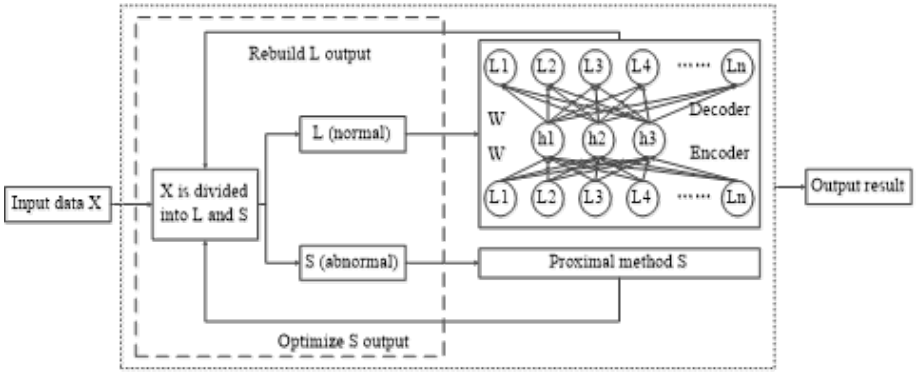


Figure 1: Deep auto-encoding model structure

Furthermore, the method of proximal optimization is used to optimize the output and then the difference between the output and the input data is calculated for the next model training. As described in PCA method, input data X is divided into two parts for processing, that is:

$$X = L + S. \tag{6}$$

Where X represents the input data, L represents the normal (clean) part of the input data. S is the abnormal data. The L is input into the auto-encoder network to effectively reconstruct the output L_D. After the input data X, $S = X - L_D$ is obtained. The output S is optimized by the proximal method, and then $L = X - S$ is obtained by subtracted from the input data X. After such alternate optimization, the model completes a training. It outputs the final result when the set number of iterations is reached. It uses PCA method to divide input data into abnormal and normal part. The model not only can be isolated from abnormal data, and the normal part is trained through the encoder, but also through the proximal optimization method, it optimizes the abnormal part, which makes iterative positive samples and negative samples be separated effectively. Finally, it achieves the effect of anomaly detection. This construction will help the model better describe the distribution of normal data in the data set, and it is expected to achieve a good effect of anomaly detection without labeled samples.

S is added to the regular part of the target function. According to formula (5) and constructed model in Figure 1, the objective function of the model can be rewritten as:

$$min_{\theta,S} = ||L_D - D_\theta(E_\theta(L))||_2 + \lambda||S||_{2,1}. \tag{7}$$

Where L is the normal (clean) data, which can be reconstructed well by the auto-encoder. S contains abnormal or noise data that is difficult to reconstruct by the auto-encoder. L_D is the output data after normal (clean) data reconstruction. E_θ and D_θ in the formula are to parameterize the encoder/decoder. λ is the parameter that adjusts the sparsity degree of S, that is, the proportion of abnormal data input in the data center. It is an important parameter in the subsequent model training. That is, if the λ is smaller, then the model considers that there are more abnormal data. On the contrary, the abnormal data is less. $||S||_{2,1}$ is the L21 norm, and its calculation formula is:

$$||S||_{2,1} = \sum_{i=1}^{m} ||x_i||_2 = \sum_{i=1}^{m} (\sum_{j=1}^{n} |x_{ij}|^2)^{0.5}. \tag{8}$$

The L21 norm can be defined that it first solves L2 norm of each row to get a vector with m row one column, and then calculates the one norm of this vector. Namely, it is the sum of L2 norm of each row in matrix X.

The output result of model $\hat{y}$ can be defined that the matrix $X - L_D - S$ calculates the L2 norm for each row. Then it obtains a column vector with m row, as shown in Equation (9).

$$\hat{y} = (\sum_{j=1}^{n} |x_{ij}|^2)^{0.5}. \tag{9}$$

Where x_{ij} is the i-th row and j-th column value in $X - L_D - S$.

Since the S matrix represents abnormal data, when a row in the S matrix is basically close to the 0 vector, it indicates that the input instance of this article is likely to be normal data, and the output result $\hat{y}$ is approximately the L2 norm of $X - L_D$. When a row vector in the S matrix cannot be ignored, it indicates that the input instance is likely to be abnormal data, and the output result $\hat{y}$ is shown in Equation (9). The decision function of the final model can set a corresponding threshold to classify the data into abnormal or normal data.

3 Proposed Detection Model

3.1 Depth Auto-encoding Algorithm

The detailed algorithm of the constructed deep auto-encoding model in this paper is shown Algorithm 1. From Step 3 to 10, they are the main DAE algorithm. Step 3 first initializes S and L_D as the 0 matrix. And it randomly initializes the weight W and bias b in the auto-encoding

network. Steps 5-10 are one training of the DAE model. Steps 7 and 8 are the reconstructed input part by the traditional auto-encoder. Steps 9 and 10 are the improved deep auto-encoder. The sparse matrix S is specifically updated according to the reconstructed L_D, and the sparse matrix is optimized by proximal optimization method. In this way, the DAE model completes a training, and when the training times reach the upper limit of the iteration number *iteration_limit*, the model outputs the final decision score S_n.

Algorithm 1 DAE:deep auto-encoding model

1: Input. $X \in R^{m\times n}$. m is the number of input data instances, and n is the dimension of each instance data.
2: Output. $S_n = \hat{y}_n(n = 1, 2, \cdots, m)$. It outputs the decision score S_n that the probability of each instance belongs to a positive sample. This instance data is normal data when the value of S_n approaches 1. Otherwise, this instance data is abnormal data when the value of S_n approaches 0.
3: Initializing $S \in R^{m\times n}$, $L_D \in R^{m\times n}$.
4: Let the initial training iteration number $t = 0$. *Iteration_limit* is set according to different data sets.
5: While ($t <$ *iteration_limit*) do
6: $L_D = X - S$.
7: Back propagation (BP) algorithm is used to minimize the objective function $||L_D - D(E(L_D))||_2$ and optimize the auto-encoding network parameter $\theta = (W, b)$.
8: $L_D = D(E(L_D))$.
9: $S = X - L_D$.
10: The sparse matrix S is optimized by the proximal optimization method. $S = prox(l_{2,1}(S), \lambda)$.
11: At the end of one model training, $t = t + 1$.
12: End
13: The decision score S_n of each instance data is calculated based on the model training results. In other words, the matrix $X - L_D - S$ calculates the L2 norm for each row. Finally, in order to better evaluate the performance of the model, the output results are normalized to the interval of $(0, 1)$. Then $S_n = minmax\_scale(||X - L_D - S||_2)$, where the L2 norm is to operate on rows.
14: The model outputs S_n.

3.2 Proximal Optimization Method

The proximal optimization method provides a proximal operator, which is equivalent to the threshold function [4]. Compared with the λ value, the S part of the data is optimized and updated. After each iteration, the S part of the normal data (positive sample) will be set to 0, while the S part of the abnormal data (negative sample) will be set to the value related to the threshold. When the 2 norm of the row is greater than λ, its optimized S value will be larger, and the model will consider that the probability of the current row is larger. Therefore, after proximal optimization algorithm, the normal data and the abnormal data can be separated effectively, so as to achieve the effect of abnormal data detection.

The proximal optimization method is used to optimize the S in step 10 of **Algorithm 1**, and the S is solved according to the λ value. Where, λ is a threshold for regulating normal data and abnormal data. If the λ is smaller, the model is more sensitive to data. The model considers that the proportion of abnormal data is higher. The proximal optimization algorithm is as Algorithm 2, where Steps 4-9 are the main part of the algorithm. Steps 5, 6 calculate the 2-norm of each row in S. If the 2-norm of the current row is greater than λ, the row will be updated; otherwise, it will be set as 0 directly. Therefore, it can be iterated through the proximal optimization algorithm every time to separate the normal data from the abnormal data, which is also an important parameter in the subsequent model parameters.

Algorithm 2 Proximal optimization method

Input. $S \in R^{m\times n}$, λ.
2: Output. S_{output}.
Initializing output matrix S_{output}.
4: for i in 1 to m
$L2_{norm_i} = (\sum_{j=1}^{n} |S[i, j]|^2)^{0.5}$
6: If $L2_{norm_i} > \lambda$.
for j in 1 to n
8: $S[i, j]_{output} = S[i, j] - \lambda \frac{s[i,j]}{L2_{norm_i}}$
Otherwise, $S[i, j]_{output} = 0$.
10: Output optimized sparse matrix S_{output} according to λ.

3.3 Model Training Method

In this paper, DAE adopts the Alternating Direction Method of Multipliers (ADMM) method for model training. ADMM divides the objective function into two or more parts to optimize the model. Firstly, some parts are optimized and others are fixed. Equation (6) is divided into two parts. One part optimizes L, so S is fixed as a constant. The optimization objective is $||L_D - D_\theta(E_\theta(L))||_2$. The other part optimizes S, so L is fixed as a constant. The first part is optimized by using the Back Propagation (BP) method, which is a classical optimization method in deep learning training model [19]. When optimizing S, minimization regular term of proximal optimization mentioned in 3.2 is used.

Finally, the training of the whole model is actually to first fix S, uses BP algorithm to train the auto-encoder and minimizes $||L_D - D_\theta(E_\theta(L))||_2$. Then it fixes L and uses the method of proximal optimization to optimize the regularization part. In each alternate minimization, $S = X - L_D$ and $L_D = X - S$ are calculated, respectively, until the iteration number is reached. Then the

model training is completed, and the final result is output. From the construction and training of the whole algorithm, it can be seen that the DAE model training does not need labeled data and belongs to unsupervised learning, which has a wider application than semi-supervised learning. Moreover, the constraint term (regular term) is added to the DAE model, which makes the model less prone to over-fitting and has stronger generalization ability, which is expected to obtain more reliable results.

4 Experiments and Analysis

The experiment is implemented in the Intel Core i7 CPU1.6GHZ, Graphics Card AMDHD8600,4G RAM environment and Windows10 operating system. Python3.5 is used for the simulation experiment. The DAE algorithm is conducted by TensorFlow framework, and other baseline algorithms are implemented by PyOD own package.

4.1 Evaluation Index

The AUC (Area Under Curve) value is used to evaluate the performance of the model [11]. TP (True Positive) refers to the correct number of predicted positive samples. FP (False Positive) is the number of prediction errors in the positive prediction samples. TN (True Negative) is the number of correct predictions in the predicted negative sample. FN (False Negative) is the number of prediction errors in negative samples.

ROC is Receiver Operating Characteristic Curve. Each point on the ROC curve reflects the sensitivity to the same signal. The area enclosed with the coordinate axis under the ROC curve is the AUC value. Where, the horizontal axis is FPR (False Positive Rate): $FPR = \frac{FP}{FP+TN}$, and the vertical axis is the True Positive Rate (Recall): $TPR = \frac{TP}{TP+FN}$. AUC value is a probability value between 0.1 and 1. The higher the AUC value is, the more likely the current classification algorithm is to rank the positive samples in front of the negative samples. It can better classify and effectively evaluate the quality of the model. Due to the inhomogeneity of positive and negative samples in abnormal detection data, the AUC value is not affected by the imbalance of positive and negative samples. Thus the experiment mainly compares the AUC value with different algorithms.

4.2 Data Set

In this paper, the data sets used in PyOD toolkit are compared with the traditional anomaly detection methods [22]. PyOD is a comprehensive and extensible Python exception detection kit, which integrates many traditional exception detection algorithm models. To verify the performance of DAE algorithm, 7 multi-dimensional point data sets are selected for comparison inn this paper. The column number represents the feature the dimension and the label value (*i.e.*, y value). The row number represents the sample number of the data set. The DAE model inputs X as the multidimensional data. The required label (y value) 1 represents the anomaly and 0 represents the normal value. Table 1 shows the detailed information of each data set. Note: the second column is the data samples number. The third column is the data dimension. The fourth column is outlier ratio.

Table 1: Data information

Dataset	Samples	Dimensions	Outliers(%)
cardio	1831	21	9.6122
mnist	7603	100	9.2069
satimage-2	5803	36	1.2235
wbc	378	30	5.5556
musk	3062	166	3.1679
optdigits	5219	64	2.8758
pendigits	6870	16	2.2707

4.3 Results Analysis

For each dataset, it is first randomly shuffled into two parts: 80% for training and 20% for testing. In order to study the influence of different super-parameters on the results of the model, the mnist data is taken as an example. The model with different λ values and auto-encoding network structures are tested to obtain the AUC values under various conditions. Then it finds the optimal super-parameters, as shown in Table 2 and Table 3. Table 2 compares the effects of different auto-encoding hidden layers and the number of neurons on the AUC value. It sets the optimal parameters for other parameters in this model. According to the experiment, when the auto-encoding hidden layer is 3 layers and the number of neurons in the first layer is between 0.6 and 0.7 of the input data dimension, the AUC value reaches to the optimal value. Table 3 analyzes the influence of λ on the model. According to Algorithm 2, λ is a threshold for regulating normal data and abnormal data. In Table 1, the outlier proportion in mnist data set is 9.21%, so the parameters of model λ are adjusted. When the determined outlier ratio by the model is close to 9.21%, then λ value is used as the optimal parameter for the model. When $\lambda = 1$, the outlier proportion is close to the actual value, and the AUC value reaches to the maximum, then λ value is the optimal parameter in the model. Note: A=Auto-encoding hidden layer network structure.

Different super-parameters are studied on mnist data set by the model, and the same method is used to fine-tune the parameters in the other 6 data sets. The optimal super-parameters are shown in Table 4. Where, the main parameters that need to fine-tune are: The number of hidden layers in the auto-encoder, the number of neurons in each hidden layer (layers), the batch size selected in a training, the learning rate of this model, and the super-parameter (λ) defined in the model.

Table 2: AUC values of different auto-encoding network structures under mnist data set

A	Layer	AUC
[64,32]	3	0.8704
[64,32,16]	3	0.8761
[64,16,8]	3	0.8821
[64,32,16,8]	4	0.8678
[64,32,16,8,4]	5	0.8701
[70,50,30]	3	0.8836

Table 3: The performance of auto-encoding model under different k values

λ	Outlier ratio (%)	AUC
0.1	100	0.5143
0.9	12.58	0.8721
0.95	10.32	0.8789
1	8.91	0.8836
1.05	6.64	0.8822
1.1	5.32	0.8714
2	1.47	0.8657
5	0.06	0.8403

At present, the commonly used methods for data anomaly detection mainly include ABOD, FB, IForest, KNN, LOF, MCD, PCA, OCSVM and AE. In here, the first seven methods are unsupervised machine learning methods, OCSVM is semi-supervised machine learning method, AE is semi-supervised deep neural network method. The DAE model proposed in this paper is an unsupervised deep neural network method. In order to prove the unsupervised anomaly detection effect of the proposed DAE model, seven unsupervised machine learning methods, one semi-supervised machine learning method and one semi-supervised deep neural network method are selected for comparative experiments.

In the experiment, the optimal super-parameters in Table 4 are adopted to draw ROC curves with the 10 methods. Figure 2 shows the ROC curves of each method under the minst data set. The AUC value of the DAE model is compared with other 9 methods. The AUC value ranking of 10 methods in each data set is investigated. The results are shown in Table 5.

Note: The format of the data in the table is AUC/RANK. RANK is the rank of the AUC values of each method in each data set. The bold is the optimal result in the current data set.

According to the table, the DAE model achieves the best AUC value in 4 data sets, the second best AUC value in one data set, and the third and sixth in the other two data sets respectively, which shows a good abnormal data detection effect. On the other hand, in the 7 unsupervised

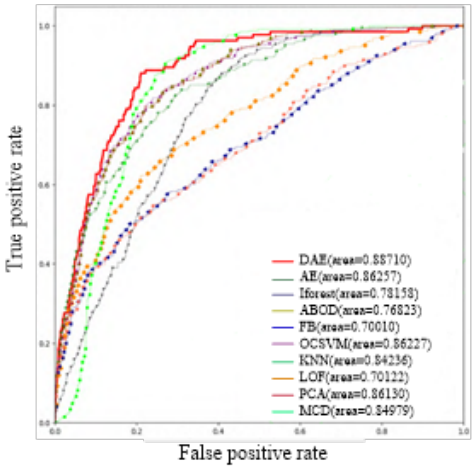


Figure 2: ROC curve with minist datase

machine learning methods, only IForest and KNN have best value one time in the wbc and the optdigits data set respectively, but the ranking of the two methods in the other 6 data sets is relatively poor. For example, the AUC value of IForest in the mnist data set (0.7965) is significantly lower than that of the DAE model (0.8836). The AUC value of KNN in the pendigits dataset (0.7487) is significantly lower than the AUC value of the DAE model (0.9532), showing a dependency on the dataset.

Although the AUC value of OCSVM method in satimage-2 and musk is better, AE method in three data sets achieves sub-optimal situation, the AUC values are obviously lower than the DAE in other data sets Also semi-supervised learning needs normal to train the model, which has certain limitations in anomaly detection for the actual data. From the above comparative test results, it can be seen that the 9 methods adopted in this experiment all have a great dependence on the data set, and the stability and reliability of anomaly detection in different data sets are poor. The DAE model has stronger stability and reliability and is more suitable for actual data anomaly detection. The improved model not only overcomes the limitation that traditional self-coding requires normal data for model training, but also has higher AUC value and model stability in data anomaly detection than traditional machine learning model and traditional auto-encoding model, which has great advantages over traditional data anomaly detection methods.

5 Conclusions

In this paper, a deep auto-encoding network model (DAE) is constructed by improving the original auto-encoding model and combining with the principal component analysis method. The proximal optimization method is used to optimize the regularization part of the objective function, the back propagation optimization method is adopted to optimize the auto-encoding part. And ADMM is used to train the whole model. In the experimental

Table 4: Super-parameter selection in deep auto-encoding model

DAE model	Layers	Iteration	λ	Batch size	Learning rate	Dropout rate	Regularizer
cardio	[15,10,5]	40	1.3	10	5.00E-05	0.01	1.00E-04
mnist	[70,50,30]	50	1	1000	5.00E-05	0.2	1.00E-04
satimage-2	[32,16,8]	50	2.7	1000	3.00E-04	0.1	1.00E-04
wbc	[24,16,10]	60	2.1	50	6.00E-05	0.1	1.00E-04
musk	[120,100,50]	60	1.5	500	1.00E-04	0.2	1.00E-04
optdigits	[51,32,19]	100	1.4	1000	8.00E-05	0.2	2.00E-01
pendigits	[10,8,5]	100	2.1	50	1.00E-05	0.2	1.00E-04

Table 5: AUC value with 10 algorithms

Model	ABOD	FB	IForest	KNN	LOF	MCD	PCA	AE	DAE
cardio	0.5693/10	0.5899/8	0.9227/5	0.7238/7	0.5737/9	0.8269/6	0.9505/3	0.9587/2	0.9593/1
mnist	0.7816/8	0.7205/9	0.7966/7	0.8482/6	0.7162/10	0.8554/3	0.8527/5	0.8657/2	0.8837/1
satimage-2	0.8190/8	0.4589/9	0.9953/3	0.9536/7	0.4577/10	0.996/2	0.9821/5	0.9823/4	0.9812/6
wbc	0.9048/9	0.9325/4	0.9268/6	0.9367/1	0.9349/2	0.9175/7	0.9159/8	0.7399/10	0.9334/3
musk	0.1844/9	0.5289/7	0.9996/4	0.7686/6	0.5287/8	0.9993/5	0.9995/3	0.9998/2	1/1
optdigits	0.4667/6	0.4437/8	0.7202/1	0.3708/10	0.4501/7	0.4034/9	0.5086/3	0.4883/5	0.5852/2
pendigits	0.6878/8	0.4676/10	0.9452/2	0.7487/7	0.4698/9	0.8332/6	0.9353/4	0.9364/3	0.9533/1

part, the deep auto-encoding model is applied to 7 real data sets and compared with 9 methods. The experimental results show that the model presented in this paper is optimal in 4 data sets and sub-optimal in 1 data set, and 3rd, 6th ranked in other two data sets, respectively. The comprehensive results of DAE are better than other methods. Therefore, as far as its performance is concerned, the new model in this paper can provide a new way for the current data anomaly detection, which has certain research value. In the future, we will study more deep learning methods and apply them into data anomaly detection.

Acknowledgments

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

References

[1] A. Alabdulatif, I. Khalil, H. Kumarage, *et al.*, "Privacy-preserving anomaly detection in the cloud for quality assured decision-making in smart cities," *Journal of Parallel & Distributed Computing*, vol. 127, pp. 209-223, 2018.

[2] R. C. Aygun, A. G. Yavuz, "Network anomaly detection with stochastically improved autoencoder based models," in *IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud'17)*, 2017. DOI: 10.1109/CSCloud.2017.39.

[3] H. Bao, Y. Wang, "A C-SVM based anomaly detection method for multi-dimensional sequence over data stream," in *IEEE International Conference on Parallel & Distributed Systems*, 2016. DOI: 10.1109/ICPADS.2016.0127.

[4] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," *Optimization*, vol. 2010, no. 2, pp. 691-717, 2015.

[5] V. L. Cao, M. Nicolau, J. McDermott, "A hybrid autoencoder and density estimation model for anomaly detection," in *International Conference on Parallel Problem Solving from Nature*, pp. 717-726, 2016.

[6] A. Dewanje and K. A. Kumar, "A new malware detection model using emerging machine learning algorithms," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 24–32, 2021.

[7] O. Dong, Y. Il, "Residual error based anomaly detection using auto-encoder in SMD machine sound," *Sensors*, vol. 18, no. 5, pp. 1308, 2018.

[8] C. Fan, F. Xiao, Y. Zhao, J. Wang, "Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data," *Applied Energy*, vol. 211, pp. 1123-1135, 2018.

[9] T. T. Gao, H. Li, and S. L. Yin, "Adaptive convolutional neural network-based information fusion for facial expression recognition," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 17-23, 2021.

[10] J. Ji, M. J. Jang, O. E. Kwon, *et al.*, "Power transmission dynamics in micro and macro slip regions for a metal v-belt continuously variable transmission under external vibrations," *International Journal of*

Automotive Technology, vol. 15, no. 7, pp. 1119-1128, 2014.

[11] S. Karim, Y. Zhang, S. Yin, M. R. Asif, "An efficient region proposal method for optical remote sensing imagery," *IEEE International Geoscience and Remote Sensing Symposium*, pp. 2455-2458, July 2018. DOI: 10.1109/IGARSS.2018.8518098.

[12] P. Li, Z. Chen, L. T. Yang, *et al.*, "An improved stacked auto-encoder for network traffic flow classification," *IEEE Network*, vol. 32, no. 6, pp. 22-27, 2018.

[13] I. Nevat, D. M. Divakaran, S. G. Nagarajan, *et al.*, "Anomaly detection and attribution in networks with temporally correlated traffic," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 131-144, 2018.

[14] M. Sabokrou, M. Fathy, M. Hosseini, "Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder," *Electronics Letters*, vol. 52, no. 13, 2016.

[15] W. Shang, L. Li, M. Wan, *et al.*, "Industrial communication intrusion detection algorithm based on improved one-class SVM," *World Congress on Industrial Control Systems Security (WCICSS'15)*, 2015. DOI: 10.1109/WCICSS.2015.7420317.

[16] G. A. Susto, A. Beghi, S. McLoone, "Anomaly Detection through on-line Isolation Forest: An application to plasma etching," *The 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO'17)*, 2017. DOI: 10.1109/ASMC.2017.7969205.

[17] E. Tonnelier, N. Baskiotis, V. Guigue, *et al.*, "Anomaly detection in smart card logs and distant evaluation with Twitter: A robust framework," *Neurocomputing*, vol. 298, pp. 109-121, 2018.

[18] Y. Wang, Z. Xie, K. Xu, *et al.*, "An efficient and effective convolutional auto-encoder extreme learning machine network for 3D feature learning," *Neurocomputing*, vol. 174, pp. 988-998, 2015.

[19] S. Yin, Y. Zhang and S. Karim, "Region search based on hybrid convolutional neural network in optical remote sensing images," *International Journal of Distributed Sensor Networks*, vol. 15, no. 5, 2019.

[20] J. Yu, H. Li, "Modified immune evolutionary algorithm for IoT big data clustering and feature extraction under cloud computing environment," *Journal of Healthcare Engineering*, vol. 6, no. 1, pp. 1-11, 2019. DOI: 10.1155/2020/1051394.

[21] Q. Zhang, C. Bai, Z. Chen, *et al.*, "Deep learning models for diagnosing spleen and stomach diseases in smart Chinese medicine with cloud computing," *Concurrency and Computation: Practice and Experience*, 2019. DOI: 10.1002/cpe.5252.

[22] Y. Zhao, Z. Nasrullah, Z. Li, "PyOD: A python toolbox for scalable outlier detection," *Submitted to The Journal of Machine Learning Research*, 2019. arXiv:1901.01588.

Biography

Rui Yang biography. Rui Yang is a lecturer in the School of Electrical Engineering & Zhengzhou University of Science and Technology. His research interests focus on computer and network security.

Dong Ye biography. Dong Ye is a lecturer in the School of Electrical Engineering & Zhengzhou University of Science and Technology. His research interests focus on computer and network security.

Digital Copyright Protection System for Oil and Gas Knowledge Achievements Based on Blockchain

Tao Feng[1], Renyi Yang[1], and Renbin Gong[2]
(Corresponding author: Tao Feng)
School of Computer and Communication, Lanzhou University of Technology[1]
287 Lan-gong-ping Road, Lanzhou, Gansu 730050, China
PetroChina Research Institute of Petroleum Exploration and Development (RIPED)[2]
20 Xue-yuan Road, Beijing 100083, China
Email:fengt@lut.cn , yry1030@163.com
(Received Jan. 13, 2020; Revised and Accepted Dec. 10, 2020; First Online May 31, 2021)

Abstract

In recent years, the infringement problem of digital copyright has become more serious. To strengthen the digital copyright protection of knowledge achievement data, this paper aims to apply the blockchain to the digital copyright protection of knowledge achievement data in the oil and gas industry. Firstly, this paper analyzes the content and format of oil and gas knowledge achievement data and designs a set of metadata structures of oil and gas knowledge achievement data. Secondly, combined with the block structure characteristics in the blockchain, the data structure of the system block is designed and implemented. Finally, according to the characteristics of the block data structure designed in the system, the relevant functional functions are designed and developed by using smart contracts. Based on the above work, the development of the system was realized by using the technology of Hyperledger Fabric, InterPlanetary File System (IPFS), and the national secret algorithm.

Keywords: Blockchain; Digital copyright protection; Data structure; Smart contracts; Hyperledger Fabric

1 Introduction

With the continuous development and advancement of global digitization, intelligence, major companies are paying more and more attention to the development of digitization, intelligence, and informatization. Since many years, various oil fields and research institutions have accumulated a large amount of knowledge and achievements in oil and gas exploration and development. The digital copyright protection of these knowledge achievement data is particularly important for the development of the oil and gas industry. In the current context of digital copyright protection, digital copyright protection is facing a series of problems such as complicated procedures, long process time, high cost and unable to provide effective legal supervision [3, 8]. Therefore, how to reduce the cost of copyright registration, ensure the legitimate rights and interests of copyright owners, and improve the supervision of the law has become an urgent problem to be solved in today's copyright protection [19, 22].

With the continuous development of blockchain technology, digital copyright protection has ushered in a new development space. The characteristics of blockchain technology, such as decentralization, distributed storage, non-tampering, security, transparency and scalability, providing a perfect and applicable solution for the digital copyright protection of knowledge achievement data, and providing a strong support for the development and reform of digital copyright protection [11].

In this paper, a digital copyright protection system of oil and gas knowledge achievement data based on consortium blockchain is designed and implemented by using blockchain. Before the system design, this article analyzed the format and content of the knowledge achievement data of the oil and gas industry. Due to the particularity and privacy of the oil and gas knowledge achievement data to a certain extent, in terms of implementing digital copyright protection with blockchain technology, this article firstly designed the metadata data structure of the oil and gas knowledge achievement data, on the premise of ensuring the security and privacy of oil and gas knowledge achievement data, it realized the secure storage and digital application of oil and gas knowledge achievement data. At the same time, according to the structural characteristics of the blocks in the blockchain, a set of block data structures for oil and gas knowledge achievements that meet system specifications and User needs are designed. Based on the above work, the smart

contract in the blockchain is used to design and implement related functional functions. The implementation of the functional functions meets the User's business needs for the system. In this system, the holder of knowledge achievement is the node of the system to manage the copyright information. At the same time, the digital copyright protection system based on blockchain also provides the User with the corresponding identity access mechanism and User rights. Only after the User passes the examination of the system can he register his identity and obtain his public and private keys. In this system, we use Inter Planetary File System (IPFS) to store knowledge achievement data, and SM2 algorithm is used to realize the hash value encryption protection of the knowledge achievement data of oil and gas, which ensures the privacy and security of the digital copyright related information of the knowledge achievement data of oil and gas.

2 Related Work

Digital rights protection technology is also known as Digital Rights Management (DRM). DRM is a new type of digital rights protection technology, which combines encryption technology, digital identification technology, identity authentication, digital watermarking and electronic transaction technology.

Recently, many scholars have developed research methods that combine blockchain technology with traditional digital rights protection technology. Ding *et al.* [6] designed and implemented three functions of copyright information protection, copyright information query and copyright information transaction by combining blockchain technology with digital copyright registration protection, and by using technologies such as DPOS consensus mechanism, ellipse cryptography, and smart contracts in blockchain technology. The design and implementation of the system meets the Users' security and credibility in the application process of digital copyright registration, storage, verification, authorization and transfer, and effectively protects the digital copyright information. However, the TPS and throughput of the system are not very high in the application process.

Xu *et al.* [21] proposed a digital rights management scheme for network media based on blockchain, in which effective production management, copyright management, transaction management and User behavior management of network media data can be achieved. However, in this solution, only the transaction information records about copyright are stored in the blockchain, and the storage and application of other copyright information are ignored. Therefore, the solution is not complete during the design process.

Zhang *et al.* [23] proposed a system solution for digital rights management based on blockchain. In this system, it stores copyright-related transaction information and some other copyright permission information in the blockchain's ledger. And through the design of smart contracts, Users can set the price information of copyright transactions themselves. Meanwhile, a more optimized license information structure was designed in the system scheme. However, there are nodes of the blockchain platform that cannot handle high concurrent key acquisition requests.

Kwame *et al.* [1] proposed a digital copyright protection system based on digital fingerprints, Inter Planetary File System (IPFS) and blockchain technology. In this system, digital media data is uploaded to IPFS for storage, and then the digital data is digitally protected. The fingerprint is recorded in the ledger of the blockchain. In this way, the system implements application functions such as on-chain storage of digital media copyright information and ensuring the integrity of digital copyright rights. However, the application function of this system is single and cannot meet the needs of multi-target Users for the protection of digital copyright information.

Xie proposed a digital copyright protection and management system with fine-grained usage control based on blockchain [10]. Through this system, Users can more purposefully, securely and conveniently manage their digital rights information. At the same time, the system provides Users with corresponding copyright information registration, digital copyright transactions, and digital content information sharing applications. The system has good performance in terms of performance and security, but the system is more suitable for the registration of digital rights information for a single User, and there are still great shortcomings and deficiencies in the negotiation of multiple User registration.

3 Background

3.1 Blockchain

As the underlying technology supporting bitcoin, people now can utilize blockchain technologies in many field and service, such as financial market, IOT, supply chain, voting, medical treatment and storage [15]. Blockchain, as a decentralized and distributed shared database, combines data blocks in a chronological order into a specific data structure. The use of cryptography ensures the security of block data [24]. As a database, blockchain uses the corresponding consensus algorithm to generate and update the ledger data information of nodes in P2P network, and uses automatic script code to program and operate data [12]. In general, the blockchain has the characteristics of decentralization, collective maintenance, programmability, immutability, security and reliability [20]. Blockchain is a kind of state machine based on peer-to-peer networks [13]. Blockchain adopts timestamp proofing, cryptography and other technologies, coupled with the distributed storage structure of the block to make the block chain decentralized and difficult to falsify forgery and collective maintenance, which ensures the security and privacy of important data in the block [9].

3.2 Hyperledger Fabric

Hyperledger Fabric is a licensed blockchain platform in the consortium blockchain, and it is also one of the super ledger projects hosted by the Linux Foundation. The project structure of fabric includes various components, such as smart contract, endorsement node, submitter, verification node and subscriber [18]. Fabric is also a modular and extensible open-source system, which is convenient to deploy and operate the alliance blockchain [2], in which consensus, network and other functional modules are developed and designed in a pluggable way. With the continuous development of fabric, the later version 1.0 and later versions can provide extensible services for application developers, and the multi-channel function [17] is adopted after the version 1.0, which solves the related business pain and enhances the security, flexibility and adaptability of the system. In the fabric architecture, modules are fully decoupled by gRPC protocol and abstract interface, and the modification in one module will not affect the function of other modules.

3.3 Inter Planetary File System (IPFS)

IPFS is a new computer technology which aims to integrate multiple technologies such as P2P networking technology, BitTorrent transmission technology, Git version control, self-certifying file system and data transmission protocol. IPFS is a global, point-to-point distributed version of the file system, the goal is to complement and replace the hypertext transfer protocol that currently governs the Internet, and connect all computing devices with the same file system together [16]. The content addressability, versioning, P2P transmission and many other features of the IPFS system provide a new type of storage method and method for large amounts of data storage. After the knowledge achievement data is uploaded to the IPFS network, IPFS will generate a unique hash value of the uploaded data according to the hash algorithm in the system. This hash value is unique in the system and it is calculated based on the content of the file of. Even if the content of the file is changed, the final hash value is completely different. The application of IPFs can not only improve the running speed and transmission speed between networks, but also provide a more secure and open data storage application mode for Users [4].

3.4 National Secret Algorithm

The National Secret Algorithm, which was a domestically-produced commercial cryptographic algorithm recognized by the National Cryptography Administration. It is a set of data encryption processing algorithms independently developed and innovated in China. In the national secret algorithm, mainly including SM1, SM2, SM3, SM4, SM7, SM9 and other cryptography algorithms. In China, the standard of the national secret algorithm is commercial cipher, which is used in various commercial fields in China. It can realize the security protection of some sensitive internal information and data that does not involve the content of state secrets. Among these algorithms, SM2 is a type of asymmetric cryptographic algorithm. Compared with the RSA algorithm, RSA is a very slow cryptosystem for long messages, it is suitable for small messages [7]. However, the SM2 algorithm has a shorter key length and a faster signature speed.

4 Detailed Design of System

4.1 System Structure

The construction of a digital copyright protection system based on the blockchain is very beneficial and practical for the protection of digital rights of knowledge achievements. The chain structure of the blockchain can completely record the entire change process of the copyright information of the knowledge achievement data. The Merkel tree structure in the block ensures that the ledger data of the blockchain cannot be tampered with. The smart contract establishes the corresponding function for the relevant business logic and behavior of the system and ensures the normal operation of the system. The decentralized feature of blockchain ensures the distributed storage of copyright information of knowledge achievement data. Combining the various aspects of the above blockchain, this paper proposes a digital copyright protection system for oil and gas knowledge achievements based on the consortium blockchain. The overall architecture is shown in Figure 1.

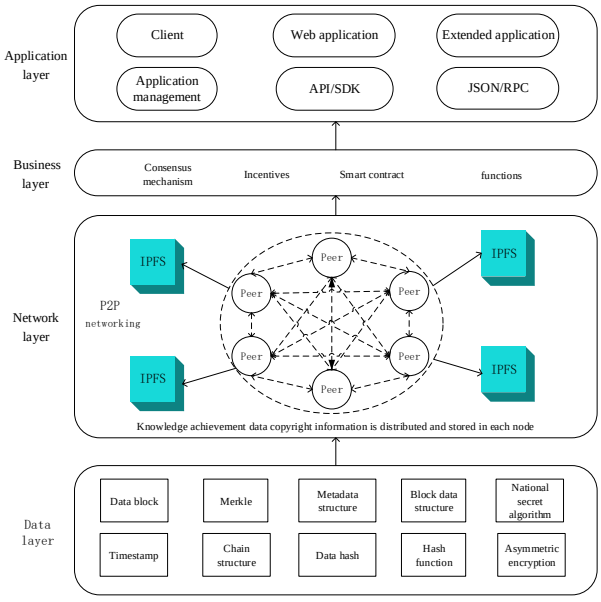


Figure 1: System architecture

As can be seen from the overall architecture of the system, the overall architecture of the blockchain based digital copyright protection system is divided into data layer, network layer, business layer and application layer.

The data layer contains the designed metadata structure of oil and gas knowledge achievements, the data structure of the block, the cryptographic method, the hash value of the oil and gas knowledge achievements data, and the data blocks, timestamps, block chain structure and other related data attribute structure information in the blockchain. The network layer contains the P2P networking mechanism and IPFS between the nodes of the system. The P2P networking mechanism provides a point-to-point network connection between network nodes, and provides the possibility for distributed storage of copyright information on oil and gas knowledge achievement data. Meanwhile, as a point-to-point distributed file system, IPFS can be used to store a large amount of oil and gas knowledge achievement data. Based on the previously designed metadata structure of oil and gas knowledge achievements and the block data structure of the blockchain, the business layer designs and realizes the relevant functions of the system, such as digital copyright registration, right confirmation, copyright information query and sharing, and packages and encapsulates the "transaction" in the system. The application layer provides Users with web applications related to the digital copyright protection of oil and gas knowledge achievements. Users call the functions designed in the business model through API + SDK to realize the User's functional application requirements for the system.

4.2 The Design of Metadata Structure

In the oil and gas industry, there are various types of oil and gas knowledge achievement data, the data types are complex, and the distribution of oil and gas knowledge achievement data is relatively scattered. There is no universal metadata structure between oil fields and research institutions. In order to protect the digital copyright and manage the oil and gas knowledge achievement data conveniently and efficiently, this paper designs the metadata structure of the whole oil and gas knowledge achievement data, just as shown in Figure 2. The attribute structure of the metadata structure is designed as follows:

Metadata Structure = ID, Author information, Data type, Data attributes, Data hash, Index address

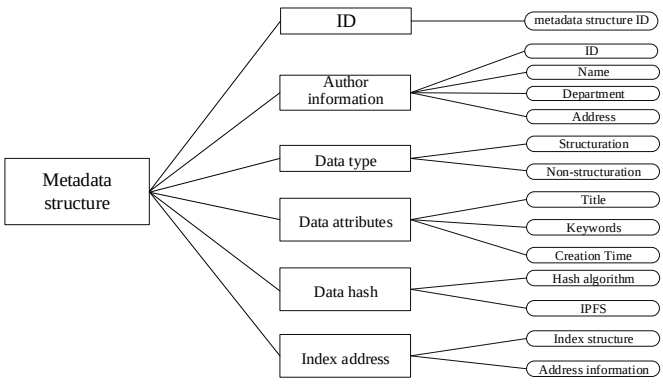


Figure 2: Metadata structure design

In the design of metadata structure attributes, Author information, Data types, Data attributes, Data hash, and Data address have their own attribute information structures, and their attribute structures are designed as shown in the following tuples.

$$\begin{aligned} Authorinformation &= \{ID, Name, Department, Address\} \\ Datatype &= \{Structuration, Non-structuration\} \\ Dataattributes &= \{Title, Keywords, CreationTime\} \\ Datahash &= \{Hashalgorithm, IPFS\} \\ Indexaddress &= \{Indexstructure, Addressinformation\} \end{aligned}$$

In the ID tuple of the metadata structure attribute, the number information of the metadata structure of oil and gas knowledge achievements is defined, and this number is the primary key query information of the metadata structure. In the author information tuple, author-related attribute information of oil and gas knowledge achievement data is defined, and this part of information includes the author's work ID, name, department, and address. In the tuple of data types, according to the types of oil and gas knowledge achievement data, the knowledge achievement data is divided into two types: structured and non-structured. In the data attributes tuple, the title represents the title information of the oil and gas knowledge achievement data, and the keywords represents the key information of the oil and gas knowledge achievement data. The creation time represents the creation and generation time of the oil and gas knowledge achievement data. In the tuple of data hash, the attribute information represents the hash algorithm used and the data hash value formed after the oil and gas knowledge achievement data is stored by IPFS. In the tuple of index address, the index structure represents the structural design for creating the address index, and the address information is the local storage information of the oil and gas knowledge achievement data.

4.3 The Design of Block Data Structure

In the chain structure of fabric, the block data structure is divided into three parts: block head, block data and block metadata. The block header contains the number of blocks, the hash value of the previous block, and the hash value of the current block. The block data includes a set of transactions that were written before the block was created. The block metadata contains information such as the block creation time, the User's certificate, public key, and signature. Therefore, based on the metadata structure of the oil and gas knowledge achievement data previously designed in this paper, the block data structure in Fabric is designed. The block data structure is shown in Figure 3.

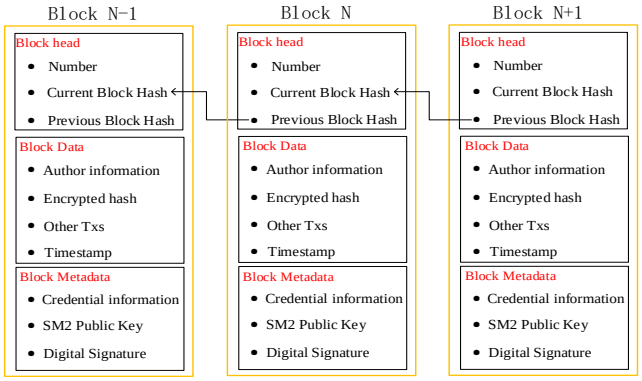


Figure 3: Block data structure

As can be seen from Figure 3, in the block data structure, block head, block data and block metadata of the block are respectively composed of attribute information of oil and gas knowledge achievement data held by Users. The data structures of the attributes are as follows:

$$\begin{aligned} BlockHeader &= Number, CurrentBlockHash, \\ &\quad PreviousBlockHash \\ BlockData &= Authorinformation, Encryptedhash, \\ &\quad OtherTxs, Timestamp \\ BlockMetadata &= Credentialinformation, \\ &\quad SM2PublicKey, DigitalSignature \end{aligned}$$

4.4 The Design of System Function

By designing the metadata structure and block data structure of the oil and gas knowledge achievement data, it provides a theoretical basis and relevant basis for the design of the system's functional functions. At the same time, based on the analysis of the requirements of the digital copyright protection system, this paper uses smart contract and other technologies to design and implement four types of functional functions: system block initialization, digital copyright registration, digital copyright confirmation, digital copyright information query and data sharing. In Table 1, we summarized the above four types of functional functions. The above four types of functional functions realize the interaction with the system by using the interface provided by the system.

Table 1: The four types of system functions

Function name	Interface	Description
initLedger	Chaincode Stubinterface	System block initialization
createDrm	Chaincode Stubinterface	Digital copyright registration
Copyright-confirmation	Chaincode Stubinterface	Digital rights confirmation
queryDrm/ queryAllDrms	Chaincode Stubinterface	Digital copyright information query and data sharing

It can be seen from the above table that createDrm function can add and register digital copyright information. The copyright-confirmation function realizes the related functions of digital Copyright confirmation, queryDrm and queryAllDrms functions realize the query of all registered digital copyright information in the system and the sharing of knowledge achievement data. The initLedger function implements the block data initialization of the digital copyright protection system. Next, we will briefly introduce and explain the business processes involved in the three types of functional functions: digital copyright registration, digital rights confirmation, and digital copyright information query and data sharing.

Digital copyright registration. The characteristics of blockchain technology, such as timestamp, P2P networking mechanism, consensus mechanism and smart contract, provide a convenient, safe and simple way for the registration of digital copyright of oil and gas knowledge achievement data. For the digital copyright registration of oil and gas knowledge achievement data, the system can pack the timestamp, the personal information of the author and the hash value of the encrypted knowledge achievement data together as a "transaction" and write them into the blockchain, so as to realize the copyright registration of knowledge achievement. The flow chart of copyright registration based on blockchain is shown in Figure 4.

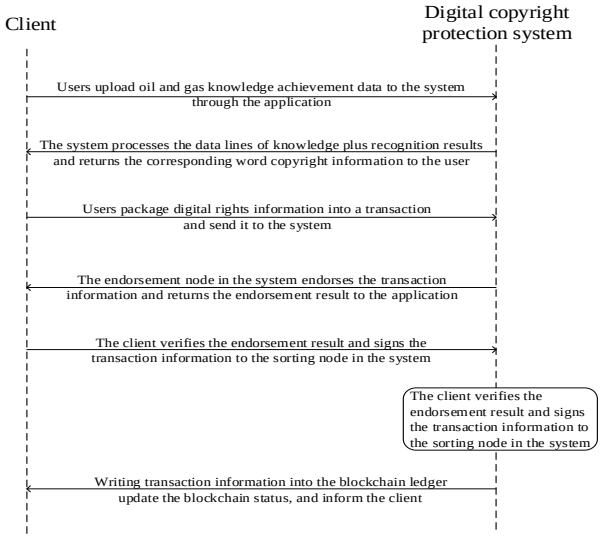


Figure 4: The flow chart of digital copyright registration

It can be seen from Figure 4 that the steps of digital rights registration based on blockchain are as follows:

1) The User uploads the oil and gas knowledge achievement data that needs digital copyright registration to the digital copyright protection system through the application client.

2) The system processes the oil and gas knowledge achievement data according to the previously designed data structure, and returns the digital

copyright information of the oil and gas knowledge achievement data to the application client.

3) The User uses the application client to package the digital copyright information of oil and gas knowledge achievement data into a transaction and sends it to the system.

4) The endorsement node in the system endorses the transaction information submitted by the client, and sends the endorsed result information to the client.

5) The client checks the endorsement result, and after passing the check, sends the transaction signature to the ordering node in the system.

6) The sorting node in the system sorts the transaction information submitted by the client, constructs the block, and sends the block to the confirmation node in the system for confirmation.

7) After the confirmation of the confirmation node is passed, the system will write the transaction information into the block, update the blockchain ledger information, and inform the client. At this point, the digital copyright registration workflow for oil and gas knowledge achievement data has ended.

Digital rights confirmation. A large number of cryptography methods are applied in blockchain technology. After Users register the digital copyright of oil and gas knowledge achievement data in the system, Users use the public and private keys provided by the system to digitally sign and verify the hash value of the encrypted knowledge achievement data. The principle and characteristics of digital signature ensure the realization of digital copyright. At the same time, the characteristics of blockchain, such as chain structure, traceability and time stamp, provide a way of legal justice for the confirmation of Users' copyright information. As shown in Figure 5, the flow chart of digital copyright confirmation is described.

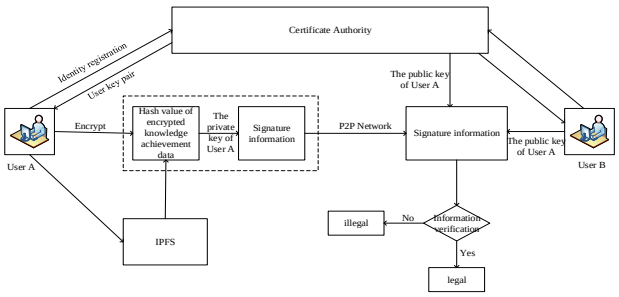


Figure 5: The flow chart of digital copyright confirmation

As can be seen from Figure 5, the whole digital signature process of digital copyright information is as follows:

1) UserA firstly uploads the oil and gas knowledge achievement data to the IPFS of the system, which will generate the hash value of the oil and gas knowledge achievement data and return the hash value to User A.

2) UserA uses the public key generated by SM2 algorithm in the system to encrypt the hash value of knowledge achievement data asymmetrically, and uses the private key to digitally sign the encrypted hash value to obtain the signature information.

3) UserA packages the hash value of the encrypted knowledge achievement data and the digital signature information together and sends it to UserB through the block chain network.

4) The Certificate Authority of the system and User B can verify the signature information by using the public key information of UserA. If the verification passes, it will prove that the digital copyright information belongs to User A. If the verification fails, it will prove that the digital copyright information of the signature does not belong to UserA.

Digital copyright information query and data sharing. Blockchain, as a database technology with features such as traceability, distributed storage, and security and transparency, provides a secure and reliable environment for querying copyright information of oil and gas knowledge achievement data and sharing of oil and gas knowledge achievement data. In the design of the system, Users in the system can query the copyright information of oil and gas knowledge achievement data that has been registered and confirmed in the blockchain ledger through this system. When the User performs the query operation, these operations will call the pre-designed function in the system to perform the relevant business logic operation for the User, and return the query result and other information to the User. If the User needs to download and apply the oil and gas knowledge achievement data according to the digital copyright information inquired, the User also needs to send a request for decryption with the holder of the knowledge achievement data. After the request passes, the User can enter the IPFS in the system to search and download the oil and gas knowledge achievement data. This implementation flow is shown in Figure 6.

It can be seen from Figure 6 that the steps of query and data sharing of the digital copyright information are as follows:

1) The holder of knowledge achievement data (User A) uses the public key PKa generated by the SM2 algorithm to encrypt the hash value of the data. The encrypted hash value will be written into the digital copyright protection system through the digital copyright registration.

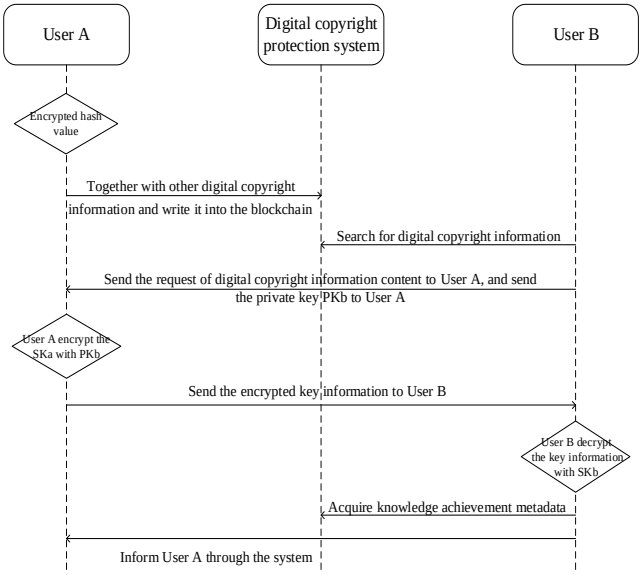


Figure 6: The flow chart of digital copyright information query and data sharing

2) The data content requester (UserB) logs in to the digital copyright protection system and obtains the digital copyright information of UserA's data content in the blockchain ledger.

3) After UserB inquires the digital copyright information of the holder of the intellectual property data, UserB sends the personal public key PKb and the hash value decryption request of the digital copyright information to UserA through the system network.

4) UserA encrypts the private key SKa generated by SM2 algorithm by using the public key PKB sent by UserB, and uploads the encrypted secret key information to the system network.

5) UserB obtains the encrypted secret key information through the digital copyright protection system network, decrypts the encrypted secret key information with its own SKb, and obtains UserA's private key SKa after decryption.

6) UserB decrypts the encrypted hash value of the digital copyright information obtained from the system by using the private key SKa, and obtains the hash value of the knowledge achievement data in IPFS by decrypting.

7) UserB can download the original data of knowledge achievements from IPFS by using hash value. After obtaining the data information, UserB will inform User A through the networks.

5 System Implementation

5.1 System Implementation

Based on the design of metadata structure, block data structure and functional functions, this system takes fabric v1.1 as the development platform. In the whole system implementation process, firstly, the network environment of fabric is built, and the image drag and source code compilation of fabric are completed by Docker and Docker-compose. Secondly, with the help of the identity management service module provided by Fabric, the system provides Users with public key, private key and certificate information. Then, the design and development of system functional functions are completed by using smart contract technology. Finally, the system interface is provided for the User, through which the User can realize a series of functions such as digital copyright registration, copyright confirmation, copyright information query and data sharing. The home page of the digital copyright protection system is shown in Figure 7. Users can register and query the copyright information of oil and gas knowledge achievements data in the home page. When Users click the link of digital copyright registration in the page, the system will jump to the digital copyright registration page for Users. At the same time, Users can realize the application functions of digital copyright confirmation, digital copyright information querying and data sharing through the system.

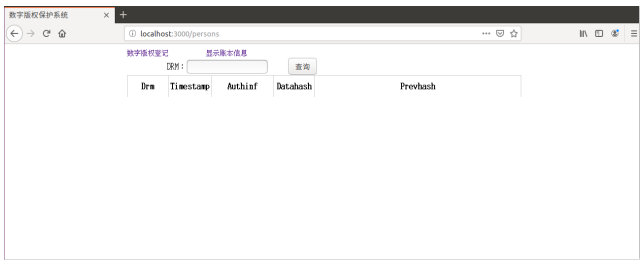


Figure 7: The home page of system

5.2 System Security Analysis

In this system, the security and privacy of digital copyright information of oil and gas knowledge achievement data are guaranteed by using IPFS, blockchain technology and related cryptographic methods. In terms of security, the system not only has a good protection ability to against traditional network security attacks, but also avoids the malicious attacks and damages of illegal users, ensuring the safe operation of the system. We will do security analysis on the system from the aspects of replay attack [5], man-in-middle attack [14], collusion attack, node failure, privacy protection and immutability.

1) Replay attack analysis. In our system, the digital copyright information exists in the form of a transaction and was stored in the blockchain ledger. Each

transaction contains information such as the corresponding number and timestamp, and each block is connected with each other through block number and block hash value.

$$\begin{aligned} & Block1 \rightarrow Block2 \\ B1 \quad = \quad & \{Noncenumber1, previoushash, \\ & randomnumber, timestamp, \\ & Merkleroot, B1hash\} \\ & Block2 \rightarrow Block1 \\ B2 \quad = \quad & \{Noncenumber2, B1hash, randomnumber, \\ & timestamp, Merkleroot, B2hash\} \end{aligned}$$

If the attacker can forge the Nonce of a block, and submit the digital copyright information and Nonce to the system as a transaction information, but during the process of writing the transaction into the blockchain ledger, the system will verify the transaction information. During the verification process, the Nonce and transaction information provided by the attacker are delayed and cannot be verified. Since the process of generating blocks for each transaction is performed in real time and is verified by the relevant nodes in the system at all times, therefore, the system can resistant replay attacks.

2) Man-in-middle attack analysis. Although the digital copyright information of the oil and gas knowledge achievement data in this system is not fully encrypted in all communication processes, in the application process of oil and gas knowledge achievement data sharing, the receiver's final received oil and gas knowledge achievement is signed by the sender. Therefore, if the oil and gas knowledge achievement data is replaced, it will not be verified at the signature verification stage. In this system, the user's identity control mechanism and session mechanism can ensure that only valid session users can access the session created by the system. Assuming that the intermediary does not send the parameter R to the user within a fixed time T, then the intermediary will not be able to complete the replacement and modification of the digital copyright information. At the same time, the Fabric platform adopted by this system adopts a multi-channel mechanism, which avoids man-in-the-middle attacks to a certain extent.

3) Anti-collusion attack analysis. This system generates a globally unique ID for each user to identify the user after registering into the system. At the same time, in the key generation stage, a random number r' and the user's ID are selected for calculation $r = H(r'|ID)$. If the hash function H used can be regarded as a uniformly determined random oracle, then all key components are unique, And the generation of key components is also random, so users cannot decrypt the ciphertext through the joint key. Therefore, the system can resist the collusion attack of users.

4) Node failure analysis. By using the distributed data storage method in this system, the single node failure problem is effectively eliminated. Different from the previous centralized storage methods, the knowledge achievement data and digital copyright information in this paper are stored in the IPFS and blockchain. These two storage methods all use distributed data storage in data storage. In this system, it is assumed that there were M nodes in both IPFS and blockchain.When N $(1 < N < M)$ nodes in the system fail or were attacked, this system can still meet the needs of users and provide corresponding application services for users. The system effectively avoids the risk that the central node of the traditional centralized data storage method is vulnerable to the paralysis of centralized malicious attacks, and avoids the paralysis of the system caused by the failure of a single node.

5) Privacy protection. In this system, many cryptography methods were used to protect the privacy of copyright information of oil and gas knowledge achievement data. In this system, a series of hash algorithms in IPFS were used to store the data of oil and gas knowledge achievements safely in IPFS. At the same time, SM2 algorithm was used to digitally sign and encrypt the copyright information of knowledge achievement data, so that the digital copyright information exists in the form of ciphertext in the blockchain of the system. This system can protect the privacy of oil and gas knowledge data and digital copyright information.

6) Immutability. In this system, all digital copyright information is stored in the blockchain. The data information on the blockchain will be arranged in chronological order, and each block will save the hash value of the previous block, assuming that the current blockchain sequence is $\{a_1, a_2, a_3, a_4 ..... a_n\}$. If the malicious node modifies a block information, assuming that the block i was modified, the malicious node also needs all the block information after modification, generating a new block $\{a_1, a_2, a_3, a_4 ... a_i, a_{i+1} ... a_n\}$, and approved by other nodes. However, for malicious nodes, they can only be recognized by other nodes after controlling more than 50% of the computing power of the whole network, which was difficult for attackers to achieve, so the digital copyright information in this system was immutability.

5.3 System Test

The local development environment of this system is virtual machine, operating system Ubuntu16.04, CPU

1.80GHz, RAM 4G. The overall core code size of the system is about 40KB, and the code size of the functional functions written by the smart contract is 11.5kb. When deploying the system and starting the network, the smart contract functions only takes 1248 milliseconds in the process of deploying and instantiating to the blockchain network node. When a single User performs a call function query to get the result, the response time of the system is about 2 milliseconds.

Next, we test the system's RPS with different transaction requests and user concurrency. In Figure 8, we show the test results. Among them, we take 100, 300, 500, 1000, 2000 and 5000 transaction requests as abscissa, and the requests per second (RPS) of the system as ordinate. Curve A represents the RPS change of the system after 10 concurrent Users execute different number of transaction requests, curve B represents the RPS change of the system after 50 concurrent Users execute different number of transaction requests, curve C represents the RPS change of the system after 100 concurrent Users execute different number of transaction requests.

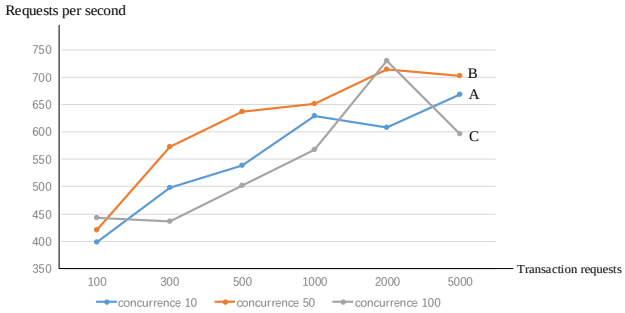


Figure 8: Performance test of the system

From the line chart of the system performance test, it can be seen that the system has a high throughput and can handle multi-User transaction requests. With the number of transaction requests increasing, the system can increase the throughput of the system within a certain range when there are fewer concurrent Users. However, with the number of Users increasing, the throughput of the system will decrease. This phenomenon is closely related to the number of concurrent Users and the number of transaction requests executed. However, on the whole, the system is easy to deploy, it takes less time to deploy, and the core code of the system is relatively simple, which is more suitable for applications between enterprises.

5.4 Scheme Analysis and Comparison

The comparison between this paper and other schemes of realizing digital copyright protection system by using blockchain technology is shown in Table 2. The application function of the system in literature [6] was relatively perfect, which can effectively protect the digital copyright information. However, the system cannot handle high concurrent user and transaction requests in the process, and the throughput of the system was low. Literature [21] designed and implemented a digital rights management system based on blockchain. In this system, only the transaction information about copyright was stored in the blockchain, and the privacy protection of the transaction information was ignored. At the same time, the application function of this system was not complete, but it can meet the high concurrent and multi-task transaction requests of users in this system, however, the throughput of the system was not low. In the scheme of literature [10], the application function of the system is single, and there are many links to be improved in the scheme. Moreover, the scheme of the system can only satisfy the common transaction request operation and application between a small number of target users in a certain range, but the system has a high throughput in the application process, and can realize the privacy protection of copyright information. In the scheme of literature [15], the system can achieve fine-grained use control and digital copyright protection and management functions for users, and realize the privacy protection of copyright information. Meanwhile, the system is more suitable for single user to operate a series of digital copyright information, therefore, the system has a high throughput in the performance test of a single user. However, the system still some defects and deficiencies in multi-user registration and transaction request processing.

Table 2: The comparisons

Scheme	Function complete	Multiple concurrent users	Multi transaction request	High throughput	Privacy protection
Literature [6]	✓	×	×	×	✓
Literature [21]	×	✓	✓	×	×
Literature [1]	×	×	×	✓	✓
Literature [10]	✓	×	×	✓	✓
Ours	✓	✓	✓	✓	✓

6 Conclusions

Nowadays, large amount of knowledge achievement data were accumulated in the oil industry. The protection of the digital copyright information of these knowledge achievement data is particularly important for the digital development of the oil industry. The traditional digital copyright protection mechanisms are based on a centralized registration model, which essentially uses trusted third-party institutions for management. In this paper, with the help of Fabric, we design a digital copyright protection system based on blockchain technology. The digital copyright protection system based on blockchain provides a kind of simple and fast decentralized function of copyright registration, right confirmation, querying and sharing for knowledge achievement data. Compared with the traditional digital copyright protection system, simultaneous interpreting of copyright information can be ensured by this system, and the operation records of copyright information can be traceable and querying.

All in all, the implementation of this system provides integrated services such as digital copyright registration, digital copyright confirmation, digital copyright information querying and sharing application for the knowledge achievement data of the petroleum industry, and provides effective solutions for a series of problems existing in the digital copyright protection, which has broad application prospects. Of course, there are some imperfections in some aspects of the system, which will be further improved and improved in the future.

Acknowledgments

This research was supported by The National Natural Science Foundation of China (No.61462060, No. 61762060) and The Network and Information Security Innovation Team of Gansu Provincial Department of Education Lanzhou University of Technology (No.2017C-05).

References

[1] O. B. O. Agyekum, Q. Xia, Y. Liu, H. Pu, and J. Gao, "Digital media copyright and content protection using IPFS and blockchain," in *Image and Graphics*, pp. 266–277, 2019.

[2] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, and Y. Manevich, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15, 2018.

[3] C. C. Chang, K. F. Hwang, M. S. Hwang, "Robust authentication scheme for protecting copyrights of images and graphics", *IEE Proceedings-Vision, Image and Signal Processing*, vol. 149, no. 1, pp. 43-50, 2002.

[4] Y. Chen, H. Li, K. Li, and J. Zhang, "An improved P2P file system scheme based on ipfs and blockchain," in *IEEE International Conference on Big Data (Big Data)*, pp. 2652–2657, 2017.

[5] S. F. Chiou, M. S. Hwang, and S. K. Chong, "A simple and secure key agreement protocol to integrate a key distribution procedure into the DSS," *International Journal of Advancements in Computing Technology*, vol. 4, no. 19, pp. 529–535, 2012.

[6] Y. Ding, H. Pu, Y. Liang, and H. Wang, "Blockchain technology in the registration and protection of digital copyright," in *International Conference on Applications and Techniques in Cyber Security and Intelligence*, pp. 608–616, 2019.

[7] M. W. D. M. G. Dissanayake, "An efficient public key cryptosystem," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 70–80, 2018.

[8] K. F. Fan, W. Mo, S. Cao, X. H. Zhao, and Q. Q. Pei, "Advances in digital rights management technology and application," *Dianzi Xuebao(Acta Electronica Sinica)*, vol. 35 no. 6, pp. 1139–1147, 2007.

[9] P. Fan, Y. Liu, J. Zhu, X. Fan, and L. Wen, "Identity management security authentication based on blockchain technologies," *International Journal of Network Security*, vol. 21, no. 6, pp. 912–917, 2019.

[10] X. Fei, *BDRM: A Blockchain-based Digital Rights Management Platform with Fine-Grained Usage Control*, 2019. Corpus ID: 199513545.

[11] W. Jian, G. Li, and Z. J. Ning, "Digital copyright protection based on blockchain technology," *Radio and Television Information*, no. 7, pp. 60–62, 2016.

[12] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *IEEE symposium on security and privacy (SP'16)*, pp. 839–858, 2016.

[13] Z. C. Li, J. H. Huang, D. Q. Gao, Y. H. Jiang, and L. Fan, "ISCP: An improved blockchain consensus protocol," *International Journal of Network Security*, vol. 21, no. 3, pp. 359–367, 2019.

[14] C. T. Li and 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, 2010.

[15] I. C. Lin and T. C. Liao, "A survey of blockchain security issues and challenges," *International Journal of Network Security*, vol. 19, no. 5, pp. 653–659, 2017.

[16] Y. Long and W. H. Wei, "Research on distributed data sharing system based on IPFS," *Internet of Things Technologies*, vol. 6, no. 6, pp. 60–62, 2016.

[17] R. Miller, "IBM unveils blockchain as a service based on open source hyperledger fabric technology," *Retrieved April*, vol. 14, pp. 2017, 2017.

[18] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'18)*, pp. 264–276, 2018.

[19] C. Y. Tsai, C. Y. Yang, I. C. Lin, and M. S. Hwang, "A survey of E-book digital right management", *International Journal of Network Security*, vol. 20, no. 5, pp. 998-1004, 2018.

[20] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[21] R. Xu, L. Zhang, H. Zhao, and Y. Peng, "Design of network media's digital rights management scheme based on blockchain technology," in *IEEE International Symposium on Autonomous Decentralized System*, pp. 128–133, 2017.

[22] Y. Y. Yan and T. Zhi, "A survey of the research on digital rights management," *Chinese Journal of Computers*, vol. 028, no. 012, pp. 1957–1968, 2005.

[23] Z. Zhang and L. Zhao, "A design of digital rights management mechanism based on blockchain technology," in *International Conference on Blockchain*, pp. 32–46, 2018.

[24] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *IEEE International Congress on Big Data (BigData Congress)*, pp. 557–564, 2017.

Biography

Tao Feng (1970-),male, born in Lintao, Gansu province, doctor, dean of school of computer and communication, Lanzhou University of Technology, researcher, doctoral supervisor, main research direction is network and information security.

Renyi Yang (1994-),male, born in Cangzhou, Hebei province, is a master's student of Lanzhou University of Technology. His current research interests include blockchain and network security.

Renbin Gong (1964-),male, born in Jingshan, Hubei province, doctoral degree, professor level senior engineer, senior technical expert of CNPC, main research direction is information engineering and internet of things.

Dynamic Pseudonym Semantic-location Privacy Protection Based on Continuous Query for Road Network

Yonglu Wang, Kaizhong Zuo, Rui Liu, and Jun Zhao
(Corresponding author: Kaizhong Zuo)

School of Computer and Information, Anhui Normal University
Anhui Provincial Key Lab of Network and Information Security, Anhui Normal University
Wuhu 241002, China
(Email: zuokz@ahnu.edu.cn)

(Received Mar. 1, 2020; Revised and Accepted Oct. 19, 2020; First Online June 23, 2021)

Abstract

Continuous queries are one of the most common queries in location-based services, although particularly useful, such queries raise serious privacy concerns. Due to frequently updated continuous locations, continuous query privacy will be disclosed. Constructing a constant anonymous user set is one of the main methods used to protect privacy information. The exiting methods don't consider query duration or users' personalized privacy requirements, and the semantic information of location isn't fully taken into account. In order to solve these problems, a dynamic pseudonym semantic-location privacy protection algorithm based on continuous query for road network is proposed in this paper. Firstly, the city road network in question is divided using the Voronoi diagram. Secondly, a secure anonymous region is generated by adopting the greedy strategy based on location and privacy requirement. Finally, the intersection of anonymous user sets at different times is calculated, and the user's identity information is updated using the dynamic pseudonym mechanism. Experimental results show that the proposed method can satisfy users' personalized privacy requirements and improve the security of privacy protection and query efficiency.

Keywords: Continuous Query; Location-based Service; Privacy Preservation; Road Network; Semantic Location

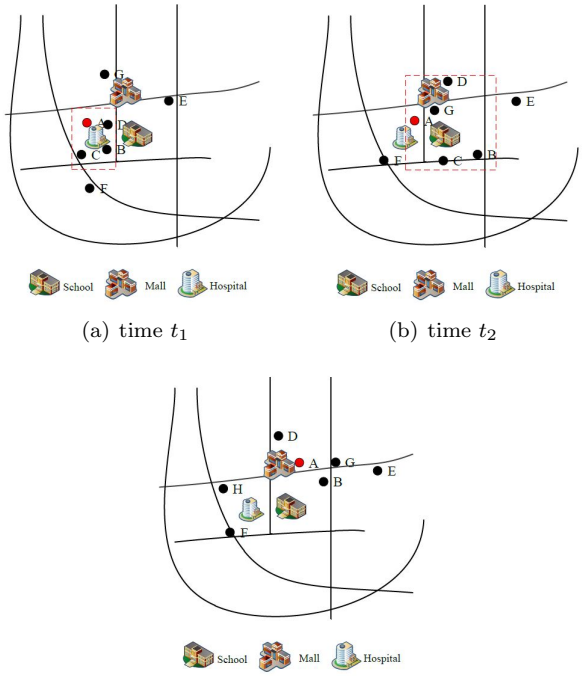
1 Introduction

With the rapid development of wireless communication technology and positioning technology, it has fuelled the wide application of location-based services (LBS) [6, 10, 12]. Continuous query as the most common LBS has become an important part of people's daily life, such as navigation and location sharing. In order to obtain services in LBS, the user has to expose their precise locations information. However, the location information can be stolen by attackers, then more private information may be leaked by the mining methods [4, 7]. Therefore, the personal privacy information about LBS should be protected [13].

K-anonymity [18] is a common method of location privacy protection, and can blur the user's exact location into spatial region, which includes $K-1$ other users. Literature [2] first pointed out that there is a query tracking attack of the continuous query, and proposed to use memorization property to build K-sharing anonymous user set. Literature [5] proposed least users and trend-based algorithm, which expands cloaking area according to the user's moving trend, and the expansion sequence of cloaking area is decided on the number of users inside cloaking area. Unfortunately, the aforementioned cloaking techniques are designed for Euclidean space, and don't consider the diversity of road segment. Based on this, K-anonymity and L-diversity has been proposed to protect location-privacy on road network, which blur the user's exact location into a set of segments. Literature [14] proposed an algorithm of continuous query privacy protection based on spatial-temporal similarity with road network. The algorithm adopts user grouping and K-sharing privacy requirement to construct anonymous user sets, and includes L road segments. Besides, Literature [8] proposed to share the same query content and time interval in the mixed region, so that the attacker can't find the real user through the sub-track, whereby protecting the location privacy. Literature [17] proposed (K, θ)-privacy model to construct an anonymous region that satisfies K-anonymity and θ-security.

However, whether it is in Euclidean space or road network, building a constant anonymous user set doesn't consider the user's query duration. As shown in Figure 1, the user A submits a continuous query service request with $K = 4$. In Figure 1(a), the dotted line is the constructed anonymous region that contains users (A, B, C, D). In

Figure 1(b), in order to deal with query tracking attacks, the anonymous region should contain users (A, B, C, D). In Figure 1(c), the user C isn't issuing the query service, so the anonymous region which contains (A, B, C, D) can't be built to achieve anonymity. Besides, it doesn't consider the semantic information of the locations. By means of the semantic location information, the attacker can infer some privacy information of the user. For example, in Figure 1(a), the attacker can infer that the user may be ill and treated in a hospital.



(a) time t_1 (b) time t_2

(c) time t_3

Figure 1: Motivation

Although there are many semantic-based location privacy protection methods [9,15,16], most of them focus on snapshot queries. If these algorithms are applied for continuous queries directly, due to query tracking attacks [2] and semantic inference attacks [9], continuous query privacy will be disclosed. Therefore, a dynamic pseudonym semantic-location privacy protection algorithm based on continuous query for road network is proposed in this paper.

The scientific contributions of this paper are summarized as follows:

1) The proposed method can meet semantic security and users' personalized privacy requirements, and fully consider query duration of the user.
2) We used the semantic information of locations, dynamic pseudonym mechanism and Voronoi diagram to construct an anonymous region.
3) We simulate our algorithm based on the real map and POIs of Beijing, compared with other algorithms, and validate the efficacy of our algorithm.

The remaining part of this paper is organized as follows. Section 2 gives definitions and system model of this paper. Section 3 describes four algorithms. Section 4 gives the experimental results and performance analysis as compared with other related methods. Finally, we conclude our paper in Section 5.

2 Preliminaries

2.1 Related Definition

Definition 1. ***Voronoi-partition road network.*** *A Voronoi-partition road network $G(V, E, Voronoi(v))$ learned from an undirected road network, $E = \{e_1, e_2, ..., e_m\}$ denotes the road segments in the road network, where each road segment $e_i = \{id, v_s, v_e\}$ is an edge in the road network, id is the road segment number, and v_s and v_e respectively denoted the start and the end point of the road segment. $V = \{v_1, v_2, ..., v_n\}$ denotes the intersection of road segment. $Voronoi(v)$ is the dominance region of the vertex v in the two-dimensional road network, that is $Voronoi(v) = \{x : d(x, v) \leq d(x, w), \forall w \neq v, (w, v) \in V\}$, where $d(x, v)$ is the Euclidean distance from x to v. Anonymous region CR is composed of multiple users $users = \{u_1, u_2, ..., u_i\}$, multiple Voronoi units $voronois = \{Voronoi(v_1), Voronoi(v_2), ..., Voronoi(v_j)\}$ and multiple semantic locations $locs = \{loc_1, loc_2, ..., loc_k\}$. A Voronoi-partition road network is shown in Figure 2.*

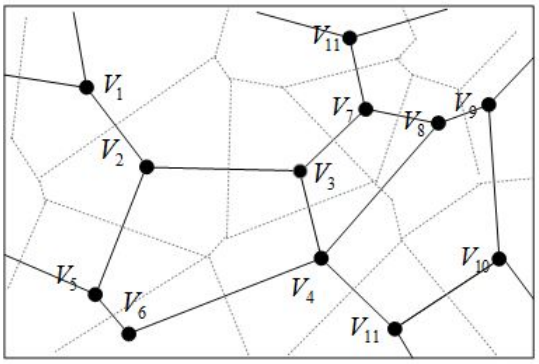


Figure 2: A Voronoi-partition road network

Definition 2. ***Semantic location.*** *$loc = \{(x, y), tp\}$ denotes the semantic location in the road network, where (x, y) is the coordinate of the semantic location and tp is the type of the semantic location. The type of semantic location is divided into n types in total, and $Type = \{tp_1, tp_2, ..., tp_n\}$ is the set of all semantic location types.*

Definition 3. ***Semantic location sensitivity.*** *It is used to describe the sensitivity of a semantic location type in the road network. Each user specifies a sensitivity score for each semantic location type according to their own circumstances. The set $SLS_{set_u} = \{sls_{tp_1}, sls_{tp_2}, ..., sls_{tp_n}\}$*

is the set of sensitivity of all semantic location types specific to user u.

Definition 4. ***Privacy requirement.*** *For a user u that makes a query, his privacy requirements are expressed in $PR(K, \theta, SLS_{set}, TN)$. In this case, K denotes user-defined the lowest number of anonymous users, θ denotes user-defined the highest value of anonymous region security; SLS_{set} is user-defined sensitivity of a group of different semantic location types, and TN denotes the user-defined number of sensitive semantic location types.*

Definition 5. ***Sensitive semantic location.*** *In the user-defined semantic location sensitivity SLS_{set}, The semantic location type corresponding to the top TN sensitivity of the largest value of the set SLS_{set} is recorded as the set MS_{set}. That is, the semantic location type belonging to the MS_{set} is the user-defined sensitive semantic location.*

Definition 6. ***Voronoi unit sensitivity.*** *The sensitivity of $Voronoi(v)$ denoted as $P_{Voronoi(v)}$ is defined by Equation (1):*

$$P_{Voronoi(v)} = \sum_{i=1}^{|Type|} sls_{tp_i} \times POP(tp_i), \quad sls_{tp_i} \in SLS_{set} \tag{1}$$

Where $|Type|$ is the size of Type, $POP_{tp_i} = \frac{|Voronoi(v).locs.tp=tp_i|}{|Voronoi(v).locs|}$. $|Voronoi(v).locs|$ represents the number of semantic locations in the Voronoi unit.

Definition 7. ***Anonymous region sensitivity.*** *The sensitivity of the anonymous region denoted as P_{CR} is defined by Equation (2):*

$$P_{CR} = \frac{\sum_{i=1}^{|MS_{set}|} sls_{ms_i} \times POP(ms_i)}{\sum_{j=1}^{|Type|} sls_{tp_j} \times POP(tp_j)}, \quad ms_i \in MS_{set}, sls_{tp_j} \in SLS_{set} \tag{2}$$

Where $|Type|$ is the size of Type, $|MS_{set}|$ is the size of MS_{set}, $POP(tp_j) = \frac{|CR.locs.tp=tp_j|}{|CR.locs|}$, $POP(ms_i) = \frac{|CR.locs.tp=ms_i|}{|CR.locs|}$. The $|CR.locs|$ represents the number of semantic locations in the anonymous region.

Definition 8. ***θ-security anonymous region.*** *If an anonymous region CR satisfies $P_{CR} \leq u.PR.\theta$, then it is called as a θ-security anonymous region.*

2.2 System Model

This paper is based on the central server architecture (Figure 3), which is the trusted third anonymous server that exists on the client and location servers. Users send their locations, inquiry contents and privacy requirements to anonymous server. The anonymous server sends the users' locations to the LBS server after anonymous modules and dynamic pseudonym mechanism are implemented. The LBS server accesses the database to obtain the candidate results and returns them to the anonymous server. The anonymous server refines the results to get the accurate results and forward them to the requester. To support this model, the anonymous server needs to store the city map information and the semantic location information.

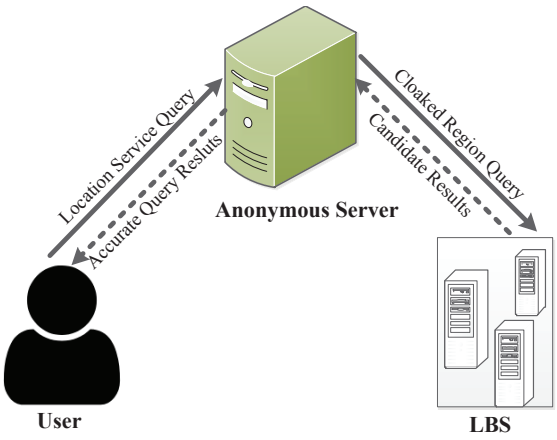


Figure 3: The proposed scheme

3 Dynamic Pseudonym Semantic-location Privacy Protection Algorithm

Given a real location of the user, we calculate the obfuscated region according to the privacy requirement PR of the user and Voronoi diagram. Algorithm 1 illustrates the details of the algorithm of obfuscated region generation. The specific steps are as follows:

Algorithm 1 Dynamic pseudonym semantic-location privacy protection algorithm (DPSPP)

Input: user u, privacy requirement PR, Voronoi-partition road network $G(V, E, Voronoi(v))$, history anonymous user set $HistoryUser_{set}$
Output: CR

1: $CR \leftarrow \varnothing$
2: Calculate MS_{set} according to PR
3: find the semantic location loc_u where user u is located
4: $CR.voronois \leftarrow CR.voronois \bigcup Voronoi(loc_u)$
5: update $CR.users$ and $CR.locs$
6: **if** $loc_u.type \in MS_{set}$ **then**
7: $CR \leftarrow Algorithm\ 2(u.PR, CR, G(V, E, Voronoi(v)))$
8: **else**
9: $CR \leftarrow Algorithm\ 3(u.PR, CR, G(V, E, Voronoi(v)))$
10: **end if**
11: **if** u is the first query **then**
12: $HistoryUser_{set} \leftarrow HistoryUser_{set} \bigcup CR.users$
13: **else**
14: $CR \leftarrow Algorithm\ 4(u.PR, CR, HistoryUser_{set})$
15: **end if**
16: return CR

(1) Initialize the input parameters;

(2) Determine whether the user's location is sensitive. If it is, construct an anonymous region according to the user's privacy requirement PR that satisfies K-anonymity and θ-security, otherwise execute Step (5);

(3) Determine whether the user is the first to submit the query. If it is, return the anonymous region CR, otherwise execute Step (4);

(4) Update the user's identity by using dynamic pseudonym mechanism and return the anonymous region CR;

(5) Construct an anonymous region CR that satisfies K-anonymity and execute Step (3).

Algorithm 2 illustrates the details of the cloaking from sensitive location. According to the CR calculated in the Algorithm 1 and user's privacy requirement, if $CR.users$ doesn't satisfy $u.PR.K$ or P_{CR} doesn't satisfy $u.PR.\theta$, we use greedy strategy to expand the minimum sensitivity in the corresponding adjacent Voronoi units until K-anonymity and θ-security are satisfied(Line 2 to 7).

Algorithm 2 Cloaking from Sensitive location

Input: $u.PR, CR, G(V, E, Voronoi(v))$
Output: θ-security CR

1: $NearVoronois_{set} \leftarrow \varnothing$
2: **while** $CR.users < u.PR.K$ *or* $P_{CR} > u.PR.\theta$ **do**
3: $NearVoronois_{set} \leftarrow FindNearVoronoi(CR)$
4: $Voronoi \leftarrow GetMinSensitivity(NearVoronois_{set})$
5: $CR.voronois \leftarrow CR.voronois \bigcup Voronoi$
6: update $CR.users$ and $CR.locs$
7: **end while**
8: return CR

Algorithm 3 illustrates the details of the cloaking from non-sensitive location. If $CR.users$ doesn't satisfy $u.PR.K$, we use greedy strategy to expand the maximum number of users in the corresponding adjacent Voronoi units until K-anonymity is satisfied (line 2 to 7).

Algorithm 3 Cloaking from Non-sensitive location

Input:$u.PR, CR, G(V, E, Voronoi(v))$
Output: CR

1: $NearVoronois_{set} \leftarrow \varnothing$
2: **while** $CR.users < u.PR.K$ **do**
3: $NearVoronois_{set} \leftarrow FindNearVoronoi(CR)$
4: $Voronoi \leftarrow GetMaxUsers(NearVoronois_{set})$
5: $CR.voronois \leftarrow CR.voronois \bigcup Voronoi$
6: update $CR.users$ and $CR.locs$
7: **end while**
8: return CR

Algorithm 4 illustrates the details of the dynamic pseudonym mechanism. If $HistoryUser_{set}$ is empty, we add $CR.users$ to $HistoryUser_{set}$ and return CR (line 2 to 4). Otherwise, the result of the intersection with $HistoryUser_{set}$ is recorded as $TempUsers$ (line 7 to 9). If $|TempUsers| < (u.PR.K/2)$, we use the pseudonym function to update the user's identity and $HistoryUser_{set} = \varnothing$ (Line 10 to 14). Otherwise, we add $CR.users$ to $HistoryUser_{set}$ and return CR (line 15 to 16).

Algorithm 4 Dynamic pseudonym algorithm

Input: $u.PR, CR, HistoryUser_{set}$
Output: CR

1: $TempUser \leftarrow \varnothing$
2: **if** $HistoryUser_{set}$ *is empty* **then**
3: $HistoryUser_{set} \leftarrow HistoryUser_{set} \bigcup CR.users$
4: return CR
5: **else**
6: $TempUser \leftarrow CR.users$
7: **for** $user \in HistoryUser_{set}$ **do**
8: $TempUser \leftarrow TempUser \bigcap user$
9: **end for**
10: **if** $|TempUser| < (u.PR.K/2)$ **then**
11: UUID(u)//Generating pseudonym
12: update $CR.users$
13: $HistoryUser_{set} \leftarrow \varnothing$
14: **end if**
15: $HistoryUser_{set} \leftarrow HistoryUser_{set} \bigcup CR.users$
16: return CR
17: **end if**

4 Experiment and Analysis

4.1 Experiment Data

(1) The experimental data is based on the Beijing map, which includes 83849 nodes and 110114 edges. There are 1000 uniform distribution users obtained from Brinkhoff based NGMO [1] by introducing the map of Beijing into Brinkhoff generator. Each user contains 100 snapshot locations, and 50 new users are added at each time. The data used for our experiment is the vector map of Beijing [3], which converts the coordinate of GCS_WGS_1984 into the coordinate of WGS_1984_UTM_Zone_50N. The specific steps are as follows:

1) Input: the coordinate of GCS_WGS_1984, such as (40.069($latitude$),116.242($longitude$))
2) $a = 6378137, e = 0.818192, k_0 = 0.9996, e_0 = 500000, N_0 = 0$
3) $ZoneNumber = 50$
4) $\lambda_0 = (ZoneNumber - 1) \times 6 - 180 + 3$
5) $\lambda_0 = \lambda_0 \times \pi/180$
6) $\varphi = latitude \times \pi/180$
7) $\lambda = longitude \times \pi/180$
8) $v = \frac{1}{\sqrt{1-e^2 \times \sin^2 \varphi}}$

9) $A = (\lambda - \lambda_0) \times \cos\varphi$

10) $T = \tan^2\varphi$

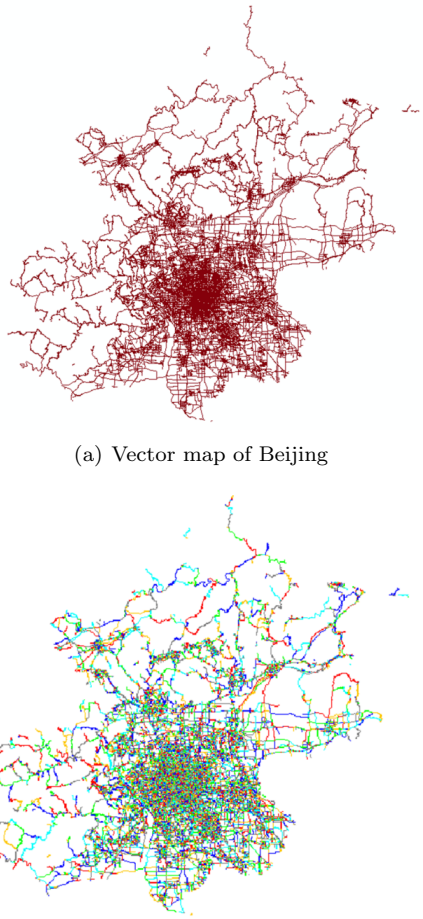
11) $C = \frac{e^2 \times \cos^2\varphi}{1-e^4}$

12) $s = (1 - \frac{e^2}{4} - \frac{3e^4}{64} - \frac{5e^6}{256}) \times \varphi + (\frac{3e^2}{8} + \frac{3e^4}{32} + \frac{45e^6}{1024}) \times \sin 2\varphi + (\frac{15e^4}{256} + \frac{45e^6}{1024}) \times \sin 4\varphi - \frac{35e^6}{3072} \times \sin 6\varphi$

13) $UTME = e_0 + k_0 \times a \times v \times (A + (1 - T + C) \times \frac{A^3}{6} + (5 - 18 \times T + T^2) \times \frac{A^5}{120})$

14) $UTMN = N_0 + k_0 \times a \times (s + v \times \tan\varphi \times (\frac{A^2}{2} + (5 - T + 9 \times C + 4 \times C^2) \times \frac{A^4}{24} + (61 - 58 \times T + T^2) \times \frac{A^6}{720}))$

15) Output: the coordinate of WGS_1984_UTM_Zone_50N, such as (435376($UTME$),44357089($UTMN$))



(a) Vector map of Beijing

(b) Map of Beijing in NGMO

Figure 4: Map of Beijing

(2) Use ShapeNetworkFileManager [11] to convert the vector map of Beijing into the Network based Generator of Moving Objects(NGMO) [1]. In order to match the map of Beijing, the map scale in NGMO is set to 130000 × 130000. The conversion formula is defined by Equations (3) and (4):

$$newX = (Input(UTME) - minX) \times res/dfX \quad (3)$$

$$newY = (maxY - Input(UTMN)) \times res/dfY \quad (4)$$

$minX$ is the minimum of abscissa in the vector map of Beijing, $minX$ = 364489.71274133306; $maxY$ is the maximum of the ordinate in the vector map of Beijing, $maxY$ = 4545213.473351848; res is the defined map scale, res = 130000; dfX is the difference between the maximum and the minimum in the abscissa, dfX = 177715.31899050274; dfY is the difference between the maximum and the minimum in the ordinate, dfY = 179208.5048302291. As shown in Figure 4.

(3) Use the Programming Interface of Baidu Map Application to mine POIs and obtain 361915 semantic locations. There are 10 types of semantic locations: Enterprise, Science and Education, Life Services, Entertainment, Hotel, Traffic, Hospital, Government, Restaurant and Financial Services.

(4) Convert the coordinates of POIs, which converts the coordinate of GCS_WGS_1984 into the coordinate of WGS_1984_UTM_Zone_50N. Then, convert the coordinates of POIs into NGMO according to Equations (3) and (4).

4.2 Parameter Settings

The environment of the experiment is Intel Core(TM) 2 CPU @2.83GHz; 2GB RAM. The operating system is Microsoft Windows 7 Professional, and the algorithm is developed in Java based on MyEclipse.

According to the POIs of Beijing and principle of the nearest distance, a specific semantic information is labeled on the attribute of location type in location data generated by Brinkhoff generator. The experiment randomly selects 10 users who request service to simulate experiments. Considering the time complexity and the quality of service, the maximum number of voronoi units V_{max} is set in the experiment. All experimental parameters in the experiment set are shown in Table 1.

4.3 Analysis of Experimental Results

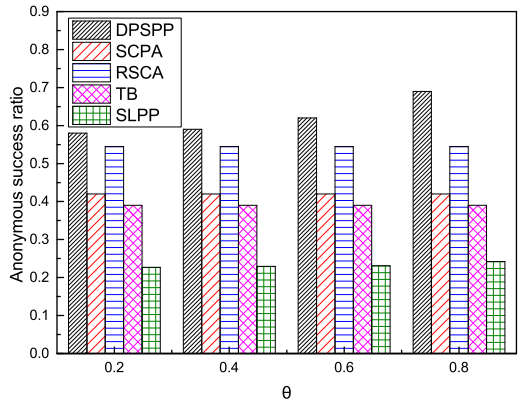
The experiment compares and evaluates our model, DP-SPP, with SCPA proposed in [14], RSCA proposed in [2], TB proposed in [5] and SLPP proposed in [17] from the aspects of anonymous success ratio, protection level and scalability.

1) Anonymous success ratio. Figure 5 depicts the performance results with respect to varying θ-security threshold level from 0.2 to 0.8, and K-anonymity level from 3 to 9. Figure 5(a) indicates that DP-SPP is higher than SCPA, TB, RSCA and SLPP. This is because that SCPA, TB, RSCA and SLPP only consider the K value and ignore query duration. Besides, the anonymous success ratio decreases when θ decreases. This is because if θ-security level is low, more non-sensitive locations need to be expanded, When the number of Voronoi units reaches

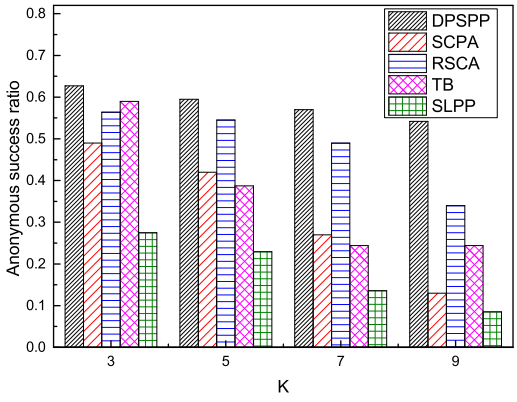
Table 1: Parameter settings

Parameters	Default values	Range
The number of users	1000	
K	5	[3,9]
The number of semantic locations	361915	
The number of users that request services	10	
V_{max}	20	
θ	0.4	[0.2,0.8]
TN	3	[1,4]
SLS_{set}	user custom	
Times	100	
New users at each time	50	

the upper limit of the Voronoi-partition, anonymity will fail. In Figure 5(b), it is clear that the anonymous success ratio of DPSPP is higher than SCPA, TB, RSCA and SLPP. This is because that DPSPP fully considers the user's continuous query duration.



(a) change of θ



(b) change of K

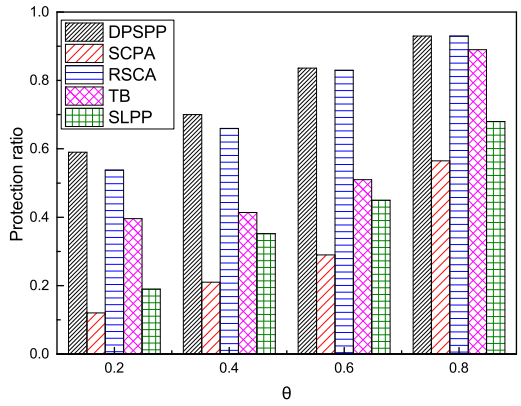
Figure 5: Anonymous success ratio

2) Protection level. This measures the level of information disclosure in sensitive-attacks regarding the information of semantic location from an anonymous region. To measure this, we introduce the average protection ratio about the number of the θ-security anonymous regions against the number of total anonymous regions. Figure 6 shows the security level to the sensitive-attack under varying θ-security level (from 0.2 to 0.8) and K-anonymity level (from 3 to 9). As shown in Figure 6, the protection ratio of SCPA, TB, RSCA and SLPP are lower than the DPSPP because that DPSPP considers the semantic-security, while SCPA, TB and RSCA doesn't. So, it is clear that our method provides better privacy protection than the other four methods under the same costs of enforcing the location privacy.

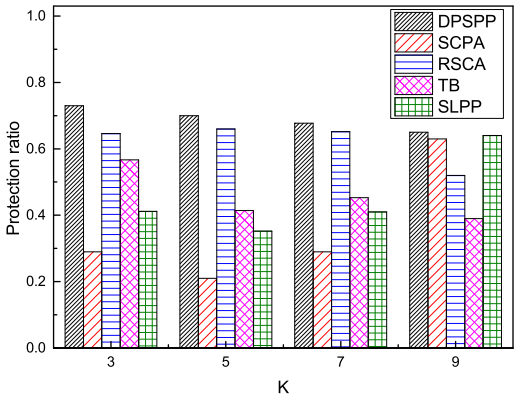
3) Scalability. This measures the efficiency performances about the increasing number of sensitive semantic locations. Figure 7 shows the success ratio, protection level, average area and average runtime with respect to varying number of sensitive semantic locations from 1 to 4. In these tests, the value of K and θ is set to 5 and 0.4 respectively. As show in Figure 7(a) and 7(b), because that SCPA, TB and RSCA don't consider semantic-security, neither of them is affected by this change. As show in Figure 7(c), DPSPP is more secure than other algorithms. As show in Figure 7(d), the execution time of DPSPP has certain growth with the increasing of the number of sensitive semantic locations. But this growth is acceptable for the scalability of our algorithm.

5 Conclusions

In this paper, we propose a dynamic pseudonym semantic-location privacy protection algorithm based on continuous query for road network. A trusted third party (TTP) is added between the users and the server for constructing anonymous region for the LBS server. The algorithm allows a user to express his/her requirement of privacy. What's more, we use semantic location information to construct an anonymous region, and design a dynamic pseudonym mechanism based on the intersection relationship of the historical anonymous user sets with a full con-

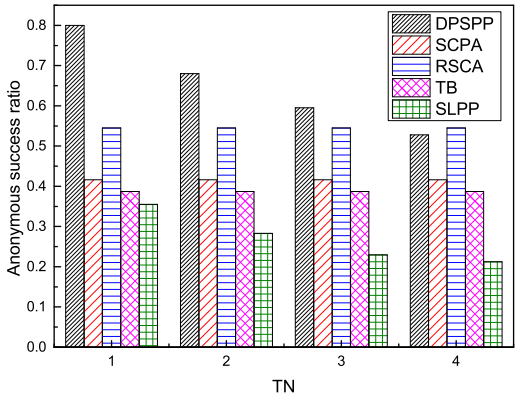


(a) change of θ

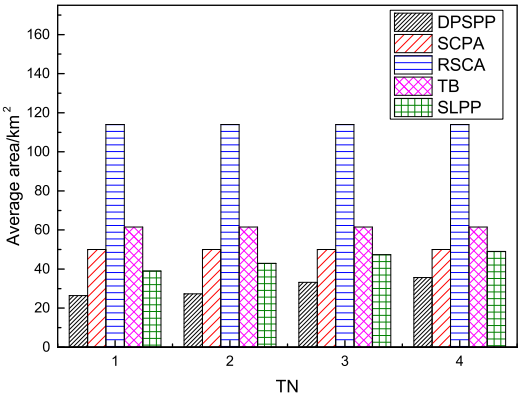


(b) change of K

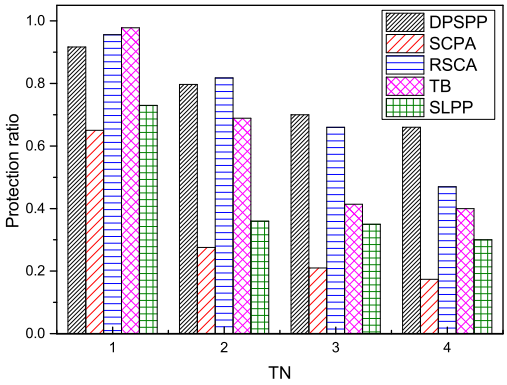
Figure 6: Protection level



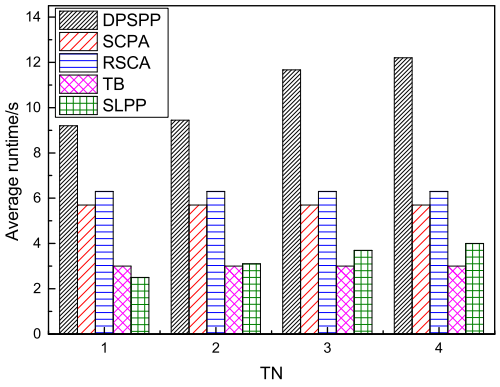
(a) anonymous success ratio for change of TN



(b) average area for change of TN



(c) protection level for change of TN



(d) average runtime for change of TN

Figure 7: Scalability

sideration of the user's query duration. However, we only consider the spatial distribution of semantic locations in this work. Therefore, how to combine the spatial distribution and the time dimension of the semantic location to enhance the degree of privacy protection is another interesting research problem we would like to investigate further in the future.

Acknowledgments

This paper is supported by the Natural Science Foundation of China through projects 61672039 and 61370050 and by the Anhui Natural Science Foundation through project 1508085QF133.

References

[1] T. Brinkhoff, "A framework for generating network-based moving objects," *Geoinformatica*, vol. 6, no. 2, pp. 153–180, 2002.

[2] C. Y. Chow and M. F. Mokbel, "Enabling private continuous queries for revealed user locations," in *International Symposium on Spatial and Temporal Databases*, pp. 258–275, 2007.

[3] Figshare, *Road Networks, Urban (2016): Urban Road Network Data*, Dataset, June 1, 2021. (`https://doi.org/10.6084/m9.figshare.2061897.v1`)

[4] S. Gambs, M. O. Killijian, M. N. del P. Cortez "Show me how you move and I will tell you who you are," *Transactions on Data Privacy*, vol. 4, no. 2, pp. 103–126, 2011.

[5] J. Gan, H. Xu, M. Xu, *et al.*, "Study on personalized location privacy protection algorithms for continuous queries in LBS," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 98–108, 2016.

[6] H. Huang, G. Gartner, J. M. Krisp, *et al.*, "Location based services: Ongoing evolution and research agenda," *Journal of Location Based Services*, vol. 12, no. 2, pp. 63–93, 2018.

[7] J. Krumm, "Semantic-aware dummy selection for location privacy preservation," in *Proceedings of the Pervasive Computing, 5th International Conference*, pp. 127–143, 2007.

[8] Z. Lei, M. Chunguang, Y. Songtao, *et al.*, "Users collaborative mix-zone to resist the query content and time interval correlation attacks," *Tehnicki Vjesnik-Technical Gazette*, vol. 25, no. 4, pp. 962–969, 2018.

[9] M. Li, Z. Qin, and C. Wang, "Sensitive semantics-aware personality cloaking on road-network environment," *International Journal of Security & Its Applications*, vol. 8, no. 1, pp. 133–146, 2014.

[10] D. Lu, Q. Han, and K. Zhang, "A novel method for location privacy protection in LBS applications," *Security and Communication Networks*, pp. 1–11, 2019.

[11] Network-based Generator of Moving Objects. (`https://iapg.jade-hs.de/personen/brinkhoff/generator/`)

[12] Y. Sun, M. Chen, L. Hu, *et al.*, "Asa: Against statistical attacks for privacy-aware users in location based service," *Future Generation Computer Systems*, vol. 70, no. 4, p. 48–58, 2017.

[13] H. J. Wu, Y. H. Chang, M. S. Hwang, I. C. Lin, "Flexible RFID location system based on artificial neural networks for medical care facilities", *ACM SIGBED Review*, vol. 6, no. 2, 2009.

[14] P. Xiao, C. Weizhang, S. Yige, *et al.*, "Continuous queries privacy protection algorithm based on spatial-temporal similarity over road networks," *Journal on Communications*, vol. 54, no. 9, pp. 2092–2101, 2017.

[15] M. Xu, H. Xu, C. Xu, *et al.*, "Personalized semantic location privacy preservation algorithm based on query processing cost optimization," in *International Conference on Security*, pp. 153–168, 2017.

[16] H. Xu, Y. Zheng, J. Zeng, *et al.*, "Location-semantic aware privacy protection algorithms for location-based services," in *IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, pp. 1219–1224, 2018.

[17] W. Yonglu, Z. Kaizhong, Z. Haiyan, *et al.*, "Sensitive-semantic location privacy protection for continuous query," *Computer Engineering and Applications*, vol. 56, no. 14, pp. 74–81, 2020.

[18] Y. B. Zhang, Q. Y. Zhang, Z. Y. Li, *et al.*, "A k-anonymous location privacy protection method of dummy based on geographical semantics," *International Journal of Network Security*, vol. 21, no. 6, pp. 937–946, 2019.

Biography

Yonglu Wang He was born in 1992. He is a Master's student at Anhui Normal University. His major research fields include data security and privacy preservation.

Kaizhong Zuo He was born in 1974. He is a professor and a supervisor of Master's student at Anhui Normal University. His major research fields include data security and privacy preservation.

Rui Liu She was born in 1995. She is a Master's student at Anhui Normal University. Her major research fields include data security and privacy preservation.

Jun Zhao He was born in 1997. He is a Master's student at Anhui Normal University. His major research fields include data security and privacy preservation.

AccountVerif: A General Framework of Verifying Accountability Protocols

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

School of Computer Science and Technology, University of Science and Technology of China
The Library of West Campus of USTC, Huang Shan Road, Hefei, Anhui Province, China
Email: huangwc@ustc.edu.cn

(Received Mar. 18, 2020; Revised and Accepted Oct. 12, 2020; First Online June 1, 2021)

Abstract

Accountability protocols need to be concretely and formally verified. The major challenge is to define a judge's responsibility: Ideally, the judge should discover every malicious operation, but in practice, dishonest participants can easily escape from the judge's surveillance. We propose a new and widely applicable formal verification framework on accountability protocols, and implement an extension of applied pi calculus, **AccountVerif**, to model and verify the protocols. We proved the generalizability of our framework by formally verifying a non-repudiation protocol, and find weaknesses in the protocol.

Keywords: Accountability; Formal Methods; Network Security; Protocol Security

1 Introduction

Accountability protocols [13,29] have been widely studied, such as elections [25], voting [11] and non-repudiation [21, 28].

In these protocols, three types of agents may be involved: the participants, the Dolev-Yao adversaries [12], and the judge. Briefly, the participants are the ones designed in the protocols, while the Dolev-Yao adversaries are the outsiders who can perform various of malicious operations. One of the desired adversarial models in accountability protocols assumes that besides Dolev-Yao adversaries, participants may also perform malicious operations. Then, an honest judge is mandated to fulfill the following duties to detect and account for the malicious operations: discovering the malicious operations and blaming the dishonest participants who have committed the operations. Since the accountability is solely defined and determined by the judge, it is critical to ensure that the decisions made by the judge are correct. Unfortunately, although formal analysis has been successful in finding the design flaws in security protocols [5, 14, 16, 18, 20, 27], few studies exist in formal modeling and verification of the accountability protocols.

This paper is to provide a general and formal framework which can be used to automatically verify the accountability protocols. The main challenge is to formally model the judge's responsibilities. Intuitively, the goals that should be met by the judge include:

- **Fairness:** No honest participants are blamed mistakenly by the judge, *i.e.,* 100% in the *precision*.
- **Completeness:** All dishonest participants are successfully blamed, *i.e.,* 100% in *recall*.

Although these tentative definitions are intuitive, they are not complete and solid definitions in the following sense. Operations that are not following the protocols are not necessarily malicious. For example, a participant who sends a harmless "Hello" message [6] is not following the protocol. Besides, the judge cannot discover all the malicious operations in practice, but the defined completeness impractically requires the judge to detect all malicious operations. For example, a participant may stealthily send his private keys to others via a private channel, which is philosophically impossible to detect.

We have formally defined fairness and completeness without previously discussed drawbacks. The definitions can be generally applied to various accountability protocols, such as non-repudiation and auction protocols. Moreover, the definitions can be satisfied by the judge in practice, and can be formally verified as well. Specifically, we do not use the notions "malicious operations" or "dishonest participants". We denote "abnormal operations" as operations that not strictly follows the protocols and "abnormal participants" as participants who may perform abnormal operations.

In the fairness definition, the judge can only blame the dishonest participants. Recall that a participant is abnormal if, for example, he sends a harmless "Hello" message, but he is not dishonest since this message does not harm to the protocol. The judge will first identify the abnormal participants, and the dishonest participants, which is a subset of the abnormal participants, are identified by the protocol designer. The protocol designer resembles

the Legislative Branch of the government and he sets the criteria on whether a participant is dishonest, while the judge resembles the Judicial Branch and only determines whether the participant is abnormal. At the end, since the judge's operation is also regulated by the protocol designer, the judge can finally blame only the dishonest participants among the abnormal participants.

For the completeness, $\mathcal{D}$**-completeness** is defined to enumerate all possible abnormal operations that should be accounted. The definition are based on the fact that the judge should blame participants in the cases: an operation in a step designed in the protocol has not been executed, whereas the operation was supposed to be executed if there were no abnormal participants or Dolev-Yao adversaries. We proved that $\mathcal{D}$-completeness contains all the above cases, and also showed the generality of the definition when studying various accountability protocols. Besides, $\mathcal{D}$-completeness was deducted and simplified such that for an n-step accountability protocol, the abnormal participants are divided into at most $n-1$ types. Therefore, at most $n-1$ disputing processes are required to be designed to ensure that all the types of abnormal participants can be blamed in the protocol.

We propose and implement **AccountVerif** [15], an extension of the applied pi-calculus, for designing and verifying accountability protocols. By using the new clauses defined in AccountVerif, only the normal participants are required to be manually modeled, and the corresponding abnormal participants are automatically translated into internal clauses verifiable by ProVerif. Furthermore, since modeling of completeness is sophisticated and error-prone, AccountVerif also automatically generates the supplementary processes. Therefore, the complexity of formally modeling and verifying the accountability protocols are greatly reduced.

The generalizability of our framework was proved by studying an accountability protocol. We illustrate the modeling and verification process by using the non-repudiation protocol [28] as a case study. We found weaknesses in the protocol.

The contributions of the paper are as follows:

1) We designed a general framework of formally modeling and verifying accountability protocols.

2) We made a novel formal definition of fairness and completeness, which can be applied to various accountability protocols, and the definition can be achieved by the judge in practice.

3) We proposed and implemented AccountVerif [15] for designers to model accountability protocols, while AccountVerif also automatically translates the models into clauses verifiable by ProVerif language.

4) We formally verified an accountability protocol using AccountVerif and found weaknesses.

$M, N ::=$	terms
a, b, c, n, t	names
x, y, z	variables
$h(M_1, ..., M_n)$	constructor application
...	
$P, Q ::=$	processes
0	null process
$P \mid Q$	parallel composition
$!P$	replication
new $n : t;\ P$	name restriction
in$(M,\ x : t);\ P$	message input
out$(M,\ N);\ P$	message output
if M **then** P **else** Q	conditional
let $x = M$ **in** P **else** Q	term evaluation
$R(M_1, ..., M_n)$	macro usage
event $e(M_1, ..., M_n);\ P$	event
...	
$\langle$declarations$\rangle ::=$	declarations
let $R(x_1 : t_1, ..., x_n : t_n) = P.$	process macros
...	

Figure 1: Syntax of the process calculus in ProVerif

The rest of the paper proceeds as follows: Section 2 introduces preliminaries of ProVerif. Section 3 gives the models of agents and adversary. Section 4 discusses the judge's ability and Section 5 gives the method of formally modeling protocols in AccountVerif. Section 6 gives a case study of using AccountVerif. In Section 7 we discuss the related work, and finally conclude in Section 8.

2 Preliminaries of ProVerif

ProVerif [7,10] is an automatic symbolic protocol verifier, which can prove the secrecy, authentication and the observational equivalence of a security protocol. It takes as input a model of the protocol in an extension of the pi calculus with cryptography, then it automatically translates the model into an internal representation by Horn clauses.

Figure 1 gives the syntax of terms, processes, and some declarations. The identifiers a, b, c, n, t and similar ones range over names, and x, y, z range over variables. A name represents atomic data, such as keys and nonces. The syntax also assumes a set of symbols for constructors and destructors. The constructors are used to build terms, and the destructors manipulate terms in processes. For example in our paper, the constructor $\mathsf{sign}(m, k_s)$ creates a signature of message m using the private key k_s; the corresponding destructor of $\mathsf{checksign}(\mathsf{sign}(m, k_s), \mathsf{pk}(k_s)) = m$ checks the signature using the public key $\mathsf{pk}(k_s)$ and returns m only when the

signature is correct.

The processes describe operations of participants in security protocols. The null process 0 does nothing. The process $P \mid Q$ is the parallel composition of P and Q. The replication $!P$ represents an unbounded number of copies of P in parallel. The name restriction **new** $n : t$; P binds name n of type t within P. The input process **in**$(M,\ x : t)$; P inputs a message x of type t on a channel M, and it executes P with x bound to the input message. The output process **out**$(M,\ N)$; P outputs the message N on the channel M and then executes P. The conditional **if** M **then** P **else** Q executes P if M is true; otherwise, it executes Q. For the term evaluation **let** $x = M$ **in** P **else** Q, if the term M does not fail, x is bound to M and the branch P is taken; otherwise, Q is taken instead. The macro $R(M_1, ..., M_n)$ takes $M_1, ..., M_n$ as arguments, and it is declared in the list of declarations as **let** $R(x_1 : t_1, ..., x_n : t_n) = P$. The macro represents the operation processes of an honest participant, who strictly follows the protocol. The event process **event** $e(M_1, ..., M_n)$; P marks the important stage reached by the protocol but does not otherwise affect any behavior, and then executes P.

In ProVerif, a protocol can be regarded as a **control flow graph**, where each operation is a node v, and the edge $v_1 \mapsto v_2$ means v_2 may be executed after v_1 is executed. A protocol **run** is a trace (or path) in the graph. The protocol designer can perform querying to check the correspondence property. For example, if **event**$(e_1) \Rightarrow$ **event**(e_2) is successfully verified, it means that the event e_2 dominates [22] the event e_1. In other words, if e_1 is executed, then e_2 has been executed. Denote the dominator of a node e as $\mathcal{D}(e)$. Then, $e_1 \in \mathcal{D}(e_2)$ in the example.

3 Agents and Adversary Model

We assume that there are 3 types of agents in the network: participants, Dolev-Yao adversaries, and the judge. The participants are the ones who are specified in the protocols, but may not strictly follow the protocols. The Dolev-Yao adversaries are not supposed to participate in the protocol. Nevertheless, they may eavesdrop the public channels or send forged messages to participants according to the channels. The judge is a trusted participant, who receives messages and detects operations which may have been made by participants or Dolev-Yao adversary. Then, he makes decision on whether a malicious operation has been performed, and, if there has been one, who performed it. Note that the protocol should prevent attacks from Dolev-Yao adversaries. In other words, the judge cannot blame the Dolev-Yao adversaries in any case, for the Dolev-Yao adversary can be anyone who has access to the channels and cannot be identified.

The Dolev-Yao adversary model assumes that the adversary controls the network. More specifically, the adversary can perform the following operations:

1) Eavesdrop any messages from the public channel;
2) Analyze and compose the messages;
3) Send the message to the public channel, pretending as a participant;
4) But he cannot perform any cryptanalysis.

   Note that in most security protocols, it assumes that only Dolev-Yao adversaries may perform malicious operations. It implicitly assumes that the participants should strictly follow the protocol. For instance, a protocol satisfying secrecy means that the adversary cannot obtain some secret message which belongs to a participant. In this case, if a participant does not strictly follow the protocol by leaking the secret, the secrecy property is no longer satisfied. Therefore, the Dolev-Yao adversary model, which was used to prove classical properties, is insufficient for accountability protocols.

   Hence, we make additional assumptions that besides the Dolev-Yao adversary, a participant may not strictly follow the protocols.

5) Eavesdrop any messages from the public channel or the private channel that he possesses;
6) Analyze and compose any received messages;
7) Send the composed messages to the public channel or the private channel that he possesses;
8) But he cannot perform any cryptanalysis.

The assumptions 5), 6), 7) can be composed to interpret most of the participants' malicious operations in the network protocols, such as colluding or false report. For the colluding, a participant may stealthily exchange secret messages with another one. For the false report, a dishonest participant may forge the proof by composing received messages, and send it to the judge to blame innocent participants.

We also extend the assumption 4) that both adversary and participants cannot perform cryptanalysis. The assumptions make it possible for automatic symbolic analysis. Some work makes stronger assumptions, *e.g.*, computational model [3, 4], which also analyzes the probability of successful cryptanalysis. We focus on the symbolic model, and the computational model is out the scope of this paper.

4 Judge's Responsibilities

In this section, we briefly analyze the judge's abilities and responsibilities. We show the challenges of formally defining malicious operations and dishonest participants. The problems motivate us to make more appropriate definitions in Section 5.

Firstly, it is philosophically impossible for the judge to detect all the operations. The judge cannot monitor

Table 1: Types of participants

Type of Participant x	1)	2)	3)
Honest / Dishonest	Honest	Honest	Dishonest
Malicious operations	No	No	Yes
Behavor	Normal	Abnormal	Abnormal
Operations by participants	–	**NFol**(x)	**NFol**(x)
Operations by judges	–	–	**JNFol**(x)

and log all the messages when the protocol runs. For example, two colluding participants may collude by transferring messages via private channels, which cannot be accessed by the judge. Furthermore, in most protocols, the judge does not log all the messages in order to reduce the network overhead. Instead, the judge may make decisions only when the malicious operations are discovered and disputed by a participant.

Secondly, it is unnecessary for the judge to account all the operations that does not strictly follow the protocol. Hence, we need a clear definition of **dishonest participants** and the corresponding **malicious operations**. Recall that a participant may not strictly follow the protocol, but does not perform any malicious operations, *e.g.*, sending a harmless "Hello" message. It is complicated to formally define the malicious operation that we should enumerate all the possible operations of a participant in each protocol step, and exclude all the harmless operations. Moreover, we should redefine the malicious operation in a new protocol intuitively.

To illustrate the problem more clearly, we divide the participants into 3 types:

1) They strictly follow the protocols;

2) They do not strictly follow protocol, but operations can be ignored by the judge;

3) They do not strictly follow protocol, but operations cannot be ignored.

Informally, the participants of Type 1) and 2) are honest, and the ones of Type 3) are **dishonest** and have performed **malicious operations** in the protocol, as shown in Table 1. Observe that it is much easier to set the boundary between the participants of Type 1) and the participants of Type 2), 3). In Section 5, we leverage the observation and further propose the appropriate formal definition of fairness and completeness.

5 Modeling in AccountVerif

In this section, we introduce AccountVerif for modeling accountability protocols, as shown in Figure 2, and illustrate how AccountVerif translates the new clauses into ProVerif language. In AccountVerif, the types of participant's behaviors are defined. Then, we make a general definition of fairness and completeness, which can be applied in various accountability protocols.

$P, Q ::=$	processes
ain(i, id, M, $x : t$); P	accounted input
aout(i, id, M, N); P	accounted output
event NFol(id)	event
event Step($i, id_f, id_t, x_1, \ldots, x_n$)	event
event Do(i, id_f, id_t)	event
...	
⟨declarations⟩ ::=	declarations
let account $R(x_1 : \mathsf{ID}, ..., x_n : t_n) = P.$	accounted macro
type ID.	type
const free $\mathsf{C_h}$: S.	bitstring
const free $\mathsf{C_d}$: S.	bitstring
const free $\mathsf{f_{att}}$: S.	bitstring
const free $\mathsf{f_{nor}}$: S.	bitstring
free $\mathsf{c_a}$: channel.	channel
free $\mathsf{c_c}$: channel.	channel
...	

Figure 2: New clauses and reserved names in AccountVerif

5.1 Behaviors of Participants

To model fairness and completeness, we find that the definition of dishonest participants or malicious operations is not required. Instead, we define **behavior** as the process performed by a participant in a protocol run, and the participants' behaviors are divided into two types (see Table 1):

(a) *Normal behavior*, which is performed by the participant who strictly follows the protocol. It corresponds to the participants of Type 1) in Section 4, and we denote them as **normal participants**.

(b) *Abnormal behavior*, which is performed by the participant who does not strictly follow the protocol. It corresponds to the participants of Type 2) and 3), and we denote them as **abnormal participants**.

For short, we denote the abnormal participants and Dolev-Yao adversaries as **abnormal agents**. If there is no abnormal agent in a protocol run, the protocol run is denoted as an **ideal protocol run**; otherwise, if there may be abnormal agents, it is denoted as a **practical protocol run**.

Definition 1 (Accounted Process Macro)**.** *In AccountVerif, a participant's behavior, which may be abnormal, is defined by using the following macro:*

$$\textbf{let account } f(id : \mathsf{ID}, x_2 : t_2, \ldots, x_n : t_n) = P. \quad (1)$$

Here, the first parameter $id : \mathsf{ID}$ in the macro is specified as the identifier of the participant, where the type ID is reserved in AccountVerif. The rest of the parameters $x_2, \ldots, x_n$ represent the **knowledge** maintained by id, such as the private key that id owns.

The accounted process macro is different from the process macro defined in ProVerif language: **let** $R(x_1 : t_1, \ldots, x_n : t_n) = P$ (see Figure 1). The *accounted process macro* can be used to define the process of a participant who may be either a normal participant or an abnormal participant. The *process macro* only defines the normal behavior of an honest participant, such as the judge, or the other trusted third parties (TTP). Note that both the accounted process macro and the process macro are needed for modeling accountability protocols.

Automatic translation of accounted processes: AccountVerif deals with the accounted process macro by automatically generating 2 process macros **let** $f_t(id : \mathsf{ID}, x_2 : t_2, \ldots, x_n : t_n) = P$, and **let** $f_f(id : \mathsf{ID}, x_2 : t_2, \ldots, x_n : t_n) = P_f$.

Here, f_t represents the participant's normal behavior where the process P in f_t is inherited from the process in f. f_f represents the abnormal behavior, and the process P_f is generated by AccountVerif according to P. Then, AccountVerif automatically generates f in translated scripts in Figure 3. Here, the participant id chooses the normal/abnormal behavior randomly for each run of the protocol. id firstly receives a message f_c of the type bitstring (*i.e.*, S) from a public channel $\mathsf{c_c}$. Then, if f_c equals a constant $\mathsf{C_h}$, id chooses the normal behavior; otherwise, if f_c equals a constant $\mathsf{C_d}$, id chooses the abnormal behavior. Since $\mathsf{c_c}$, $\mathsf{C_h}$ and $\mathsf{C_d}$ are public, the adversary can send $\mathsf{C_h}$ and $\mathsf{C_d}$ to the channel $\mathsf{c_c}$. Hence, either f_t or f_f can be executed. The names $\mathsf{c_c}$, $\mathsf{C_h}$ and $\mathsf{C_d}$ are reserved in the AccountVerif for avoiding duplicated definition by protocol designers.

let $f(id : \mathsf{ID}, x_2 : t_2, ..., x_n : t_n) =$
  in$(\mathsf{c_c}, f_c : \mathsf{S})$;
  if $f_c = \mathsf{C_h}$
    then $f_t(id, x_2, ..., x_n)$
  else if $f_c = \mathsf{C_d}$
    then (**event** **NFol**(id); $f_f(id, x_2, ..., x_n)$).

Figure 3: Translated process macro from the accounted process macro by AccountVerif

The participant id executes the event **NFol**(id) before executing f_f, if $\mathsf{C_d}$ is received by id. We use the event to model the fairness in accountability protocols, as illustrated in Section 5.2. The event name **NFol** is also reserved in AccountVerif.

In the process f_f, we leverage the ability of Dolev-Yao adversary to model id's abnormal behavior. Briefly, to maximize id's abilities of performing malicious behaviors, id sends his knowledge to the Dolev-Yao adversary, and then relays all the messages between the adversary and participants with whom he is supposed to communicate. Moreover, to deal with the limitation of ProVerif when proving completeness (see Section 5.3), we also propose a strategy which adds more processes in f_f and we also remodel the Dolev-Yao adversaries. Since the modeling is non-trivial and error-prone for protocol designers, they are automatically generated by AccountVerif. We also show the concrete process of the translation in the Appendix [15].

5.2 Fairness

We define 2 events for modeling fairness: **NFol**(x) and **JNFol**(x). As mentioned, if the participant x chooses the abnormal behavior in a protocol run, the event **NFol**(x) is executed before x performs any other operations. If the judge believes that x has chosen the abnormal behavior, the event **JNFol**(x) is executed.

Definition 2 (Fairness)**.** *The judge can correctly blame the abnormal participant.*

$$\forall x \in \mathsf{ID}.\ \textbf{event}(\textbf{JNFol}(x)) \Rightarrow \textbf{event}(\textbf{NFol}(x)) \quad (2)$$

Here, whenever the judge executes the event **JNFol**(x), the participant x must have performed the operation not specified in the protocol.

*For an operation performed by x, the **protocol designer** determines whether x is **dishonest**, and the **judge** only decides whether x is an **abnormal participant**.* Recall the operation that x sends a harmless "Hello" message [19]. The protocol designer determines whether to execute **JNFol**(x) when the operation is detected. In verification, we only need to ensure that x chose the abnormal behavior before **JNFol**(x) is executed. The advantage is that the formal definition of fairness is general for the definition of abnormal behavior is clear. Moreover, it is flexible that the protocol designer determines whether an operation is malicious. Even if it is designed to execute **JNFol**(x) when the judge detects the "Hello" message, it only infers that the protocol is stricter, but it is also fair because x is abnormal. In other words, the protocol designer is similar to the Legislative Branch of the government. He defines the malicious operation and the dishonest participant in the protocol. The judge, who is similar to the Judicial Branch, only needs to blame the right participant according to the protocol made by the designer.

5.3 Completeness

The formal definition of completeness is much more challenging than the definition of fairness. Recall that it is philosophically impossible to detect all the abnormal operations performed by the participants. The protocol designer, who resembles the Legislative Branch, needs to specify a subset of abnormal operations as malicious operations for accounting. A practical definition proposed

by [19] is to set **goals** of the protocols, and then account all the operations that breach the goals. However, the goals for each protocol are different which depend on the context of the protocols, and they are not straightforward to be formally modeled. For example, in the auction protocol (*i.e.*, the PRST protocol) as illustrated in [19], there are three sophisticated goals which contain constraints on some specific protocol steps. Moreover, it is unknown whether all the cases that break the goals are successfully enumerated by the protocol designer. In the PRST protocol, 9 cases that break goals are listed. In the improved version of PRST protocol in [19], 10 cases are listed. Though it is manually proved in [19] that the completeness is guaranteed, as there may be many cases that breach the goals, a general and formal definition of completeness is required.

Our formal definition of completeness is more general. Informally, it is based on the **observation** that the protocol runs in an unexpected trace in case: *an operation has not been executed in a protocol run, whereas the operation was supposed to be executed if the run is ideal.* In this case, the judge should blame some participants. For example in Figure 4.a, the ideal protocol run is a trace of process $[A(i),\ B(i),\ C(i)]$, where i is the parameter representing the content of a message. Then, we show 3 typical cases that need to be accounted in Figure 4: **(b)** the missing event $B(i)$, **(c)** event disorder, where $B(i)$ has not been executed before $C(i)$, and **(d)** invalid parameters, where $A(j)$ should have been executed before $B(j)$. Note that the case **(e)** needs not to be accounted, though it is similar to the case (b). The reason is that the missing event $C(i)$ can be executed afterwards. The case **(f)**, where the additional event $D(i)$ is executed, (*e.g.*, the event of sending Hello message), needs not to be accounted as well.

To formally model and verify completeness, new clauses in AccountVerif are introduced to define processes in an ideal protocol run (Section 5.3.1). Then, we briefly illustrate how the clauses are translated by AccountVerif, after which the abnormal behaviors of participants are generated (Section 5.3.2). We make deductions by using the translated clauses and finally define $\mathcal{D}$-completeness (Section 5.3.3). We also prove the consistency between our observation and $\mathcal{D}$-completeness in the Appendix [15].

5.3.1 Clauses for Modeling Completeness

Definition 3 (Accounted Input/Output)**.** *An **accounted input/output** is a message input/output process which should be executed in an **ideal protocol run**. In AccountVerif, the accounted input/output is described as follows:*

$$\mathbf{ain}(i,\ id,\ M,\ x:t) \tag{3}$$

$$\mathbf{aout}(i,\ id,\ M,\ N) \tag{4}$$

Here, the processes **ain** and **aout** in Equations (3) and (4) resembles the definition of processes **in** and **out** respectively in Figure 1. The difference is that the step number $i \in \mathbb{N}$ and the identifier id are added as parameters. The parameter i together with id means the input/output process is executed by the participant id in the ith step of the protocol.

Therefore, 4 types of processes (*i.e.*, **in**, **out**, **ain**, **aout**) are used for modeling input and output processes. The process **ain**, **aout** are used for modeling processes in an ideal protocol run, while the process **in**, **out** are used for modeling disputes performed by participants and judges (see the example in Section 6).

Definition 4 (Accounted Step)**.** *An **accounted step** is an event process executed in a **practical protocol run**, which is derived from a pair:* $\mathbf{ain}(i,\ id_t,\ M,\ x:t)$ *and* $\mathbf{aout}(i,\ id_f,\ M,\ N)$*. It is automatically generated by AccountVerif, and has the form as follows:*

$$\mathbf{event}\ \mathsf{Step}(i, id_f, id_t, x_1, \ldots, x_n) \tag{5}$$

Here, the accounted step Step indicates that a message, which is supposed to be sent from id_f to id_t by **aout** and received by the corresponding **ain**, is sent and can be successfully validated by the processes (*e.g.*, **if** process) after the **ain** process. The parameters $x_1, \ldots, x_n$ represent the variable values in the messages that are validated, denoted as **dependent parameters**.

The accounted step is the key process for modeling and verifying completeness. In a practical protocol run, besides the normal participants, the abnormal agents can also execute the accounted steps by sending a message. In some cases, if the message sent by the abnormal agents has the same format and values as the ones in the ideal protocol runs (*i.e.*, the corresponding accounted step is executed), the operations should not be blamed. However, the message can be forged with different values which can still pass the validation after the execution of **ain**. *In this case, an accounted step has not been executed, while the next accounted steps may be executed, which should be accounted by the judge.* We informally introduce how AccountVerif generates the accounted steps for all types of participants in Section 5.3.2, and provided the formal representation in the Appendix [15].

5.3.2 Code Generation of the Accounted Steps

AccountVerif translates the accounted input and output processes into clauses in ProVerif, where the accounted steps are generated.

Case 1: Abnormal agents. In translation, AccountVerif guarantees that the accounted step is executed whenever the message N in corresponding process $\mathbf{aout}(i,\ id,\ M,\ N)$ is sent via channel M. Since the Dolev-Yao adversary in ProVerif is implicitly modeled, *i.e.*, messages in the public channel are free to be sent by Dolev-Yao adversaries, we cannot implicitly encode **Step** for these messages. Though ProVerif provides the event process **mess**() for indicating that a message is sent, using the event for verifying a correspondence property often results in

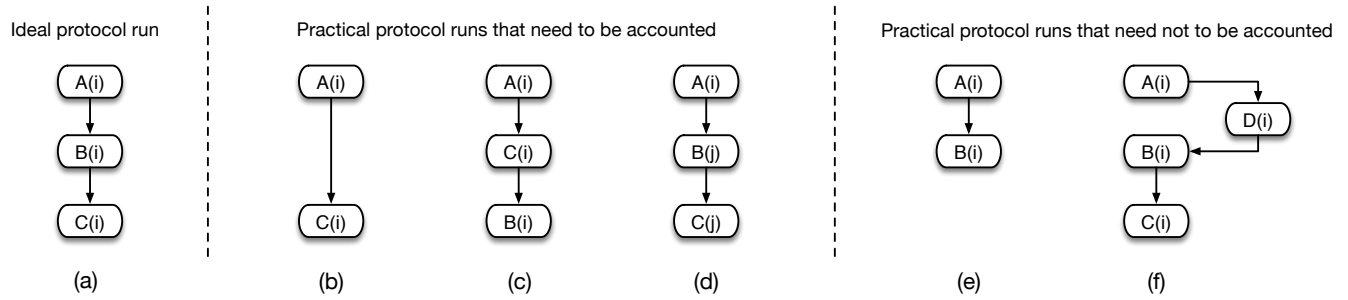


Figure 4: Example of the observation for modeling completeness: (a) Ideal protocol run. (b)(c)(d) Cases that need to be accounted. (e)(f) Cases that need not to be accounted

failure of automatic verification, due to the incompleteness problem of ProVerif. Instead, AccountVerif explicitly models the Dolev-Yao adversaries by defining a new process **att**, which resembles f_f in Figure 3, where the adversary relays all the messages between a public channel to the other channels. Moreover, the public channels defined in AccountVerif are translated into private channels. Hence, the contents of all the messages sent in **att** and f_f are examined: if an output process has the same channel and message specified in an **aout** process, the corresponding event **Step** is executed before the execution of the output process.

We make optimization based on the observation that f_f and **att** may perform the same abnormal operations. Specifically, if they can send forged messages to the same channel, *i.e.*, the channel is public, such replaying operation can be modeled only in the process **att**. If the channel defined in AccountVerif is private, since the adversary cannot access the channel, the relaying process is modeled in f_f.

Besides, we deal with the non-termination problem of ProVerif when the relaying process is modeled. Specifically, ProVerif may not terminate when performing automatic verification, if we model the relay process as follows:

$$(\mathbf{in}(\mathsf{c_a}, m_1{:}\mathsf{S});\ \ \mathbf{out}(\mathsf{c_{ab}}, m_1))|(\mathbf{in}(\mathsf{c_{ab}}, m_2{:}\mathsf{S});\ \mathbf{out}(\mathsf{c_a}, m_2))$$

Here, the abnormal agent relays messages from channel $\mathsf{c_{ab}}$ to $\mathsf{c_a}$. The reason of non-termination is that the above process would cause loops [7] in the practical runs. For example, a message is relayed from $\mathsf{c_{ab}}$ to $\mathsf{c_a}$, and then relayed from c_a to $\mathsf{c_{ab}}$, where the same message is relayed and so on. To deal with the problem, we add the reserved constants $\mathsf{f_{att}}$ and $\mathsf{f_{nor}}$, and model the replaying process as follows:

$$\begin{aligned}&(\mathbf{in}(\mathsf{c_a}, (=\mathsf{f_{att}}, m_1{:}\mathsf{S}));\ \ \mathbf{out}(\mathsf{c_{ab}}, (\mathsf{f_{att}}, m_1)))|\\ &\ (\mathbf{in}(\mathsf{c_{ab}}, (=\mathsf{f_{nor}}, m_2{:}\mathsf{S}));\ \mathbf{out}(\mathsf{c_a}, (\mathsf{f_{nor}}, m_2)))\end{aligned}$$

Here, only the message tagged with $\mathsf{f_{att}}$ (*i.e.*, generated by abnormal agents) can be relayed from $\mathsf{c_a}$ to $\mathsf{c_{ab}}$ and only the message tagged with $\mathsf{f_{nor}}$ (*i.e.*, generated by normal participants) can be relayed from $\mathsf{c_{ab}}$ to $\mathsf{c_a}$. Hence, there is no loop between relaying processes, if there is no loop in ideal runs of the protocol.

Case 2: Normal participants. These participants are modeled in the macro process or the accounted macro process (f_t) (see Figure 3). In this case, the process $\mathbf{ain}(i,\ id,\ M,\ x:t)$ is directly translated into $\mathbf{in}(M, (x_0 : \mathsf{S}, x : t))$. As already mentioned, the parameter x_0 is generated only for helping termination in relaying processes, thus AccountVerif does not generate any other process that contains x_0. Similarly, the process $\mathbf{aout}(i,\ id,\ M,\ N)$ is translated into $\mathbf{event}\ \mathsf{Step}(i, id, id_t, x_1, \ldots, x_n);\ \mathbf{out}(M,\ N')$. Here, the parameter id_t is derived from the second parameter in **ain** which has the same step i. The output message $N' = (\mathsf{f_{nor}}, N)$, thus the message N can be relayed by abnormal agents. Note that a message may be broadcast to multiple receivers, *i.e.*, there are multiple accounted input processes at the same step. In this case, AccountVerif generates multiple accounted steps when translating.

5.3.3 Modeling and Verifying Completeness

We first propose the basic mechanism of discovering cases that abnormal participants have involved, and then show how to leverage the mechanism to model and verify completeness.

Finding abnormal participants: The protocol designer uses the following event defined in AccountVerif for modeling and verifying completeness.

$$\mathbf{event}\ \mathsf{Do}(i, id_f, id_t) \tag{6}$$

The event is equivalent to the definition of accounted step, where the dependent parameters are not included. Here, AccountVerif translates the event Do into the event Step, where the dependent parameters are automatically generated. Hence, the protocol designer can perform the following queries without considering the sophisticated dependent parameters.

$$\textbf{query}\ id_1 \in \mathsf{ID}, id_2 \in \mathsf{ID}, id_3 \in \mathsf{ID}, id_4 \in \mathsf{ID}, i \in \mathbb{N}, j \in \mathbb{N}; \mathsf{Do}(i, id_1, id_2) \Rightarrow \mathsf{Do}(j, id_3, id_4) \quad (7)$$

Here, we assume $i > j$ and **event**(Do()) is abbreviated as Do(). Denote $d_i = \mathsf{Do}(i, \_, \_)$, which means d_i is an accounted step at step i. Equation (7) can be abbreviated as :

$$\forall d_i, d_j.i > j \rightarrow d_i \in \mathcal{D}(d_j) \quad (8)$$

The result of Equation (7) may be false due to the operations performed by abnormal participants. Therefore, the protocol designer can find the abnormal participants by reviewing the code and make new queries. For example, if id_3 is found to be the abnormal participant, the fact is confirmed by the following query:

$$\textbf{query}\ id_1 \in \mathsf{ID}, id_2 \in \mathsf{ID}, id_3 \in \mathsf{ID}, id_4 \in \mathsf{ID}, i \in \mathbb{N}, j \in \mathbb{N}; \mathsf{Do}(i, id_1, id_2) \Rightarrow \mathsf{Do}(j, id_3, id_4) \vee \textbf{NFol}(id_3) \quad (9)$$

Here, if d_i has been executed but d_j has not been executed, then id_3 is abnormal. According to our observation in Section 5.3, id_3 should be blamed.

As mentioned, AccountVerif translates Do in Equations (7) and (9) into Step before verifying the query. For example, Equation (7) is initially translated as follows:

$$\textbf{query}\ id_1 \in \mathsf{ID}, id_2 \in \mathsf{ID}, id_3 \in \mathsf{ID}, id_4 \in \mathsf{ID}, i \in \mathbb{N}, j \in \mathbb{N}, x_1 \in t_{x_1}, \ldots, x_m \in t_{x_1}, y_1 \in t_{y_1}, \ldots, y_n \in t_{y_n}; (\mathsf{Step}(i, id_1, id_2, x_1, \ldots, x_m) \Rightarrow \mathsf{Step}(j, id_3, id_4, y_1, \ldots, y_n) \wedge y_{t_1} = c_{t_1} \wedge y_{t_2} = c_{t_2} \ldots y_{t_k} = c_{t_k}) \quad (10)$$

The dependent parameters $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$ may share the same parameters, because the contents of the sent message in a step depends on the ones in the previous steps, as illustrated in Section 5.3.1. We informally illustrate how AccountVerif identifies the shared parameters.

Firstly, AccountVerif constructs a global map to indicate the relationship among variables (and constructors) used in the processes. Since in an ideal protocol run, a variable may be assigned to another variable according to the clauses **ain**, **aout**, *etc.*, these variables have the same values. AccountVerif builds a map $\mathcal{F}$ that for variables (or constructors) v_1, v_2 with the same value, $\mathcal{F}(v_1) = \mathcal{F}(v_2)$. Secondly, AccountVerif resolves all the variables and constructors of the messages in each **aout** process, and set them as dependent parameters in Step. Finally, in Equation (10), AccountVerif searches x_i and y_j in $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$ respectively, in which if $\mathcal{F}(x_i) = \mathcal{F}(y_j)$, AccountVerif replaces y_j with x_i.

Besides, a participant may also perform abnormal operations by changing the parameters which is supposed to be originated from his knowledge, when he executes the step j. For example, a participant B_1 may collude with B_2 by sending the messages using B_2's knowledge.

Table 2: Cases that should be accounted in protocol runs. (Yes: every accounted step at the given step is executed; No: not every accounted step is executed; -: arbitrary.)

#	d_1	d_2	d_3	d_4	d_5	...	d_{n-1}	d_n
1	No	Yes	Yes	Yes	Yes	...	Yes	Yes
2	-	No	Yes	Yes	Yes	...	Yes	Yes
3	-	-	No	Yes	Yes	...	Yes	Yes
4	-	-	-	No	Yes	...	Yes	Yes
5	-	-	-	-	No	...	Yes	Yes
...	...	...	...	...	...	...	...	...
$n-1$	-	-	-	-	-	...	No	Yes

To deal with the problem, the protocol designer declares the key data in the knowledge as private constants (*e.g.*, c_{t_1}, c_{t_k}), and Equation (10) also checks the equivalence between the constants and the corresponding parameters (*e.g.*, y_{t_1}, y_{t_k}, where $\{y_{t_1}, \ldots, y_{t_k}\} \subseteq \{y_1, \ldots, y_n\}$). Note that we do not treat all the knowledge of a participant as key data, since some data may merely be a random value and does not do harm to the protocol. The strategy of choosing the key data can be determined by the protocol designer, who resembles the Legislative Branch.

Enumerating abnormal participants: According to Equations (7) to (10) and the proof in the Appendix [15], the observation in Section 5.3 can be formally defined as:

Definition 5 ($\mathcal{D}$-completeness-init)**.** *At least one participant should be blamed, if:*

$$\exists d_i \exists d_j.i > j \wedge d_i \notin \mathcal{D}(d_j) \quad (11)$$

Therefore, the abnormal participant can be found by applying d_i and d_j to Equations (9) and (10).

Definition 5 requires to be improved due to the complexity of enumerating all the cases that some participants should be blamed. For example, if there are n steps in an ideal protocol run, $\frac{n(n-1)}{2}$ cases need to be modeled, *i.e.*, new processes for disputing and judging should be coded in each case.

To reduce the complexity of modeling, we find that the number of cases can be reduced from $\frac{n(n-1)}{2}$ to $n-1$ as shown in Table 2. For example, consider a case that d_4 is executed and d_2 has not been executed. The case can be divided into sub-cases, and allocated to the second and third row in Table 2. Therefore, each row corresponds to a new case: in the ith row, d_{n-i} is not executed, and all the following steps $d_{n-i+1}, \ldots, d_n$ are executed in sequence. Specifically, the number of cases is reduced by define the completeness as follows:

Definition 6 ($\mathcal{D}$-completeness)**.** *In case j, where $0 < j < n$ and n is the number of steps in an ideal protocol run, at least one participant should be blamed, if:*

$$\exists d_j \forall d_i.i > j \rightarrow d_i \notin \mathcal{D}(d_j) \quad (12)$$

According to Definition 6, all participants will not blamed in some cases:

Corollary 1 ($\mathcal{D}$-completeness-noblame)**.** *In case j, where $0 < j < n$ and n is the number of steps, no participants can be blamed, if:*

$$\forall d_j \forall d_i . i = j + 1 \rightarrow d_i \in \mathcal{D}(d_j) \tag{13}$$

In practice, we make the more concrete solution as follows:

Corollary 2 ($\mathcal{D}$-completeness-practical)**.** *In case j, where $0 < j < n$ and n is the number of steps, new processes can be designed for the judges to discover and blame abnormal participants, if there exist d_i, d_j which satisfy:*

1) $i > j \wedge d_i \notin \mathcal{D}(d_j) \ \wedge \ \forall d_k . j < k < i \rightarrow d_k \notin \mathcal{D}(d_j)$,

2) The channels used by d_i and d_j can be accessed by a normal participant,

3) At least one abnormal participant that causes the condition 2) can be individually found,

4) Not all the participants, who may perform malicious behaviors, are abnormal.

The condition 2) guarantees that the runs that satisfy condition 1) can be observed by a participant who can then dispute the finding. Here, d_i can be observed by a participant only if the participant can access the channel to which the message in d_i is sent.

In condition 3), the notion "individual" is derived from the definition "individual accountability" in [19], which means that at least one participant is abnormal for all the runs that satisfy condition 1). Therefore, our definition enforces individual accountability that at least one participant is responsible for each case that should be accounted. If the condition 3) cannot be satisfied for all i that satisfies the condition 1), the protocol designer should redesign the protocol to achieve individual accountability.

In condition 4), it is meaningless for the judge to blame someone, if all the participants are abnormal in practice. Note that the participants who only perform the normal behavior are not included in the participants, *e.g.*, the judge or the *TTP*.

5.4 Modeling and Verifying Process

In the end, the modeling and verification of an accountability protocol consists of the following steps:

Modeling the ideal protocol run. The designer additionally uses **ain**, **aout** and the accounted process macros to model the ideal protocol run.

Generating and verifying $\mathcal{D}$-completeness. AccountVerif translates the model and generates abnormal behaviors of participants according the accounted process macros. It also generates the accounted steps **Do** according to **ain** and **aout**. Then $\mathcal{D}$-completeness is generated according to the accounted steps and the reserved event **NFol**, and the abnormal participants are found and automatically verified by ProVerif.

Modeling the disputing cases and verifying fairness. The disputing cases are generated according to $\mathcal{D}$-completeness by using the ordinary **in**, **out** clauses, and the fairness in the disputing cases is verified by ProVerif.

Abbreviations

$$\begin{aligned} c =& m_k \\ NRO =& \{f_{nro}, B, L, c\}_{sk_A} \\ NRR =& \{f_{nrr}, A, L, c\}_{sk_B} \\ sub_k =& \{f_{sub}, B, L, k\}_{sk_A} \\ con_k =& \{f_{con}, A, B, L, k\}_{sk_{TTP}} \end{aligned}$$

Steps

$$\begin{aligned} &S_1.\ A \rightarrow B : && f_{nro}, B, L, c, NRO \\ &S_2.\ B \rightarrow A : && f_{nrr}, A, L, NRR \\ &S_3.\ A \rightarrow TTP : && f_{sub}, B, L, k, sub_k \\ &S_4.\ B \leftarrow TTP : && f_{con}, A, B, L, k, con_k \\ &S_4'.\ A \leftarrow TTP : && f_{con}, A, B, L, k, con_k \end{aligned}$$

Figure 5: The non-repudiation protocol

6 Case Study

We show a case study of formal analyzing a non-repudiation protocol [28] by using AccountVerif. We illustrate how AccountVerif translates the scripts into the ProVerif language. Furthermore, we also find weaknesses of the protocol according to the verification result.

6.1 Summary

The goal of the non-repudiation protocol is that when a participant A sends a message m to another participant B, both A and B hold the evidence that the other participant has participated in the protocol.

The brief process of the protocol is shown in Figure 5. In step S_1, A picks the symmetric key k and encrypts the message m with k to form $c = m_k$, and sends c with A's signature NRO to B. Here, f_*, such as f_{nro}, is the message flag representing the type of a message. L is the label identifying the session between A and B. Each message is also attached with the signature (*e.g.*, NRO, NRR) of customized contents using the participant's private keys (*e.g.*, sk_A, sk_B). In step S_2, B receives A's message, verifies NRO and replies to A telling that B has received c. In step S_3, A confirms the validity of the receipt and sends

the symmetric key k with A's signature to the Trusted Third Party (TTP). In steps S_4 and S_4', if TTP confirms that signature is valid, he sends the symmetric key k with his signature con_k to both A and B according to a reliable channel, which ensures that A and B receive the message. Hence, B will hold the message c and the key k, and get the decrypted message m. Moreover, A holds the evidence NRR and con_k that B has received both c and k, and B holds the evidence NRO and con_k that A has sent the message c and k. As a result, both A and B cannot repudiate the session of the protocol. Here, Both A and B may be abnormal, and TTP is normal.

6.2 Modeling Participant B in AccountVerif

The ideal protocol run contains three participants' processes (A, B, TTP). Since A, B may be abnormal, their normal behaviors are modeled by using the accounted process macros in Definition 1, by which the abnormal behaviors are automatically generated by AccountVerif. For TTP always behaves honestly, his normal behavior is modeled by using the process macro in Figure 1.

To illustrate the use of AccountVerif, we show the model of B's normal behavior in Figure 6, and how B's normal behavior, abnormal behavior and Dolev-Yao adversary's process are automatically generated into ProVerif language.

let account $P_B(B\colon \mathsf{ID}, sk_B\colon\mathsf{SK}, pk_{ttp}\colon \mathsf{PK}, pk_A\colon \mathsf{PK}) =$
  ain$(1, B, \mathsf{c_{ab}}, (=\mathsf{f_{nro}}, = B, L\colon \mathsf{S}, c\colon \mathsf{S}, NRO\colon \mathsf{S}, = pk_A)\ )$;
  if $(\mathsf{checksign}(NRO, pk_A) = (\mathsf{f_{nro}}, B, L, c)\ )$ **then**
    let $A = \mathsf{fhost}(pk_A)$ **in**
    let $NRR = \mathsf{sign}((\mathsf{f_{nrr}}, A, L, c), sk_B)$ **in**
    aout$(2, B, \mathsf{c_{ab}}, (\mathsf{f_{nrr}}, A, L, NRR))$;
    ain$(4, B, \mathsf{c_{ab}}, (=\mathsf{f_{con}}, = A, = B, = L, k\colon\mathsf{K}, con_k\colon\mathsf{S}))$;
    if $\mathsf{checksign}(con_k, pk_{ttp}) = (\mathsf{f_{con}}, A, B, L, k\ )$ **then**
      let $e_4 = \mathsf{sign}(\ (\mathsf{f_{err4}}, A, B, L, k, con_k), sk_B)$ **in**
        aout$(5, B, \mathsf{c_{ab}}, (\mathsf{f_{pnro}}, A, B, L, c, k, NRO, con_k))$.

Figure 6: The modeling of B in ideal protocol runs using AccountVerif.

In Figure 6, the parameters represent B's knowledge, including the identity B, B's private signature key sk_B, TTP and A's public signature key (pk_{ttp}, pk_B). In step S_1, B receives the message from public channel $\mathsf{c_{ab}}$ by the process **ain**. B verifies the signature NRO by the destructor $\mathsf{checksign}$, generates the signature NRR and sends the message with NRR to the channel $\mathsf{c_{ab}}$ in step S_2 by the process **aout**. Then, B receives the message from the channel c_{ftp} in step S_4 by the process **ain**. If the message is from the same initiator A and has the same session (*i.e.*, L), B verifies the signature. Finally, B sends the message to the judge proving that A has sent the message c and k (step S_5). Hence, the judge holds the evidence that B has participated in a session with A.

In P_B, the identity B is bound to B's public signature key pk_B, for it is assumed that each participant has one private signature key. It is implemented by declaring the constructor **fun** $\mathsf{fhost}(\mathsf{PK})$: ID and the corresponding destructor **reduc forall** x: PK; $\mathsf{getkey}(\mathsf{fhost}(x)) = x$. In practice, the binding can also be implemented by using the clause **table** in ProVerif. Nevertheless, the internal implementation of **table** is complicated which may result in non-termination when automatic verification is performed. Thus, we use the implementation in several protocols.

6.2.1 Automatic Code Generation by AccountVerif

The accounted macro process **let account** $P_B = \ldots$ is translated into a macro process **let** $P_B = \ldots$ by AccountVerif. The new process is the modeled in Figure 3, which contains two branches: the normal behavior P_{B_t} and the abnormal behavior P_{B_f}. Moreover, new codes are generated in the process of Dolev-Yao adversary **att**.

Generation of P_{B_t}.

In P_{B_t} the accounted steps are generated according to the **aout** processes. For example, the **aout** process in step S_2 is translated into two processes: **event** $\mathsf{Step}(1, A, B, L, c)$; **out**$(\mathsf{c_{ab}}, (\mathsf{f_{nor}}, (\mathsf{f_{nrr}}, A, L, NRR)))$. The **ain** process in step S_1 is translated into **in**$(\mathsf{c_{ab}}, (f\colon\mathsf{S}, (=\mathsf{f_{nro}}, = B, L\colon\mathsf{S}, c\colon\mathsf{S}, NRO\colon\mathsf{S}, = pk_A)))$. The tag $\mathsf{f_{nor}}$ is added for helping termination in automatic verification.

Generation of P_{B_f}. The process simply outputs all his knowledge to the public channel a as follows:

let $P_{B_f}(B\colon \mathsf{ID}, sk_B\colon\mathsf{SK}, pk_{ttp}\colon \mathsf{PK}, pk_A\colon \mathsf{PK}) =$
  out$(\mathsf{c_a}, (B, sk_B, pk_{ttp}, pk_A))$

Here, for the relaying process, since B communicates with A and TTP via the public channel $\mathsf{c_{ab}}$, the relaying process between $\mathsf{c_{ab}}$ and $\mathsf{c_a}$ is only modeled in **att** for reducing the overhead of automatic verification.

Generation of att. The process **att** overhears the public channels (*e.g.*, $\mathsf{c_{ab}}$) designed in AccountVerif, and sends forged messages to the channels. In the sending process, the accounted step is also executed, if the sent message passes the validation (see Equation (5)). For the paper size limitation, we only show generated code of the first accounted step as follows:

let att$(pk_B\colon \mathsf{PK}, sk_{ttp}\colon \mathsf{SK}, k\colon \mathsf{K}, sk_A\colon \mathsf{SK},$
    $m\colon \mathsf{S}, sk_B\colon \mathsf{SK}, pk_A\colon \mathsf{PK}, pk_{ttp}\colon \mathsf{PK}, B\colon \mathsf{ID}, A\colon \mathsf{ID}) =$
  in$(\mathsf{c_a}, (=\mathsf{f_{att}}, m_1\colon\mathsf{S}))$;
...
    let $(=\mathsf{f_{nro}}\ , = B\ , L\colon \mathsf{S}\ , c\colon \mathsf{S}\ , NRO\colon \mathsf{S}\ , = pk_A\ ) = m_1$ **in**
    (**let** $(=\mathsf{f_{nro}}\ , = B, = L\ , = c\ ) = \mathsf{checksign}(NRO, pk_A)$ **in**
    event $\mathsf{Step}(1, A, B, L, c)$;
    out$(\mathsf{c_{ab}}, (\mathsf{f_{att}}, m_1))$
) **else**
...

Here, the adversary first receives a message m_1 from c_a, which indicates that m_1 is from the Dolev-Yao adversary. Second, the message's format and its signature *NRO* are checked. The validation process is derived from B's normal behavior in Figure 6, where B checks the message sent in step S_1 before B sends the message is step S_2. Third, if m_1 passes all the validations, the corresponding accounted step is executed.

6.3 Modeling and Verifying Completeness

Table 3 shows all the cases that may involve disputing in the protocol runs. In each case, two accounted steps are chosen for modeling the completeness, and the chosen abnormal participants are automatically verified according to Equations (7) and (9). We brief illustrates the result of each case as follows:

Case 1: B may send a message in step S_2 by forging a message c in the signature *NRR*, where c is not the message received in step S_1. According to Corollary 2, in condition 1), $i = 2$, $j = 1$, and $d_2 \notin \mathcal{D}(d_1)$. In condition 2), both d_1 and d_2 can be observed by A. In condition 3), the participant B can be individually found. Therefore, A can dispute to the judge and blame B.

Case 2: A may send a message in step S_3 without having built the session L with B in step S_1 and S_2, *i.e.*, step S_2 has not been executed. According to Corollary 2, in condition 1), $i = 4$, $j = 2$, and $d_4 \notin \mathcal{D}(d_2)$, $d_3 \notin \mathcal{D}(d_2)$. In condition 2), both d_2 and d_4 can be observed by B. In condition 3), the participant A can be individually found. Therefore, B can dispute to the judge and blame A. Note that we do not choose $i = 3$, since d_2 and d_3 cannot be observed by the same normal participant (*e.g.*, B), which violates the condition 2).

Case 3: Since *TTP* is normal, d_4 is executed only if d_3 is executed. According to Corollary 1, $i = 4$, $j = 3$, and $d_4 \in \mathcal{D}(d_3)$. Therefore, no participants can be blamed in this case.

Case 4: Similar to case 3, this case also satisfies Corollary 1, and no participants can be blamed.

Hence, the designer needs to add the disputing processes for judges to find and blame abnormal participants in case 2 and 3. Note that in the original design of the protocol, the case 2 and 3 are not considered, *i.e.*, $\mathcal{D}$-completeness is not achieved.

7 Related Work

Bella *et al.* [6] formally verified a subset of accountability protocols. The validity of evidence was modeled and verified in the fair non-repudiation protocol [28] and the certified email protocol [1], but the fairness and completeness were not defined. In particular, the disputing cases in the protocols were not considered, while we found that there were cases that the participants may blame each other, and the judge has to decide who was dishonest. Küsters *et al.* [19] proposed a more general definition of accountability. They also studied the properties of several protocols by pen-and-paper analysis, *e.g.*, the Bingo voting system [8]. Nevertheless, we found that it was difficult to use the definition in formal verification. One problem is that we do not have a general definition of the notion "malicious operation" and "dishonest participant" for different protocols. Furthermore, the lacks of general and formal definition of completeness makes it challenging for verifying the accountability protocols.

Table 3: Cases that should be accounted in the non-repudiation protocol

#	d_1	d_2	d_3	d_4	d_5	Abnormal Participants
1	No	Yes				B
2	-	No		Yes		A
3	-	-	No	Yes		None
4	-	-	-	No	Yes	None

The incompleteness problem exists in many efficient tools such as ProVerif, AVISPA [26], and it is still under active research. The limitation is that the global state cannot be modeled and verified due to the abstraction from applied pi calculus to horn clauses. To solve the problem, The AIF [23] framework first presented the idea of encoding the set memberships into the state abstraction, which was based on the low-level AVISPA Intermediate Format [26]. StatVerif [2] and SetPi [9] also provided an extension of the applied pi calculus. Kremer *et al.* [17] proposed a variant of the applied pi calculus, which used the tamarin prover [24] as a backend. It supported both the global state and verification of correspondence properties, which are required in verifying accountability protocols. Nevertheless, tamarin sometimes required additional typing lemmas to guide the proof. We provided a lightweight solution of bypassing the incompleteness problem when proving fairness, and the protocols were accountability protocols verified by using AccountVerif. An interesting future work is verifying accountability protocols using the novel verification tools that supports global states.

8 Conclusion

We propose and implement AccountVerif for formally modeling and verifying accountability protocols. To reduce the complexity of modeling accountability protocols, AccountVerif extends the applied pi calculus and automatically translates the calculus into ProVerif language, where the abnormal operations and supplementary codes for modeling completeness are automatically generated. In particular, our definition of completeness is general

that it can cover all the disputing processes in existing accountability protocols, such as a non-repudiation protocol. We also find weaknesses in the protocol according to our definition.

Acknowledgments

We thank the anonymous reviewers for providing us valuable feedback for improving our paper. The research is supported by the National Key R&D Program of China 2018YFB0803400, 2018YFB2100300, National Natural Science Foundation of China under Grant No.61972369, No.61572453, No.61520106007, No.61572454, and the Fundamental Research Funds for the Central Universities, No. WK2150110009.

References

[1] M. Abadi and N. Glew, "Certified email with a light on-line trusted third party: Design and implementation," in *WWW*, pp. 387–395, 2002.

[2] M. Arapinis, E. Ritter, and M. D. Ryan, "Statverif: Verification of stateful processes," in *CSF*, pp. 33–47, 2011.

[3] G. Bana and H. Comon-Lundh, "A computationally complete symbolic attacker for equivalence properties," in *CCS*, pp. 609–620, 2014.

[4] G. Bana, K. Hasebe, and M. Okada, "Computationally complete symbolic attacker and key exchange," in *CCS*, pp. 1231–1246, 2013.

[5] M. Bayat and M. Aref, "An attribute based key agreement protocol resilient to KCI attack," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 10–20, 2015.

[6] G. Bella and L. C. Paulson, "Accountability protocols: Formalized and verified," *ACM Transactions on Privacy and Security*, vol. 9, no. 2, pp. 138–161, 2006.

[7] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in *Proceedings of the 14th IEEE Computer Security Foundations Workshop*, pp. 82–96, 2001.

[8] J. M. Bohli, J. Müller-Quade, and S. Röhrich, "Bingo voting: Secure and coercion-free voting using a trusted random number generator," *IACR Cryptology ePrint Archive*, vol. 2007, pp. 162, 2007.

[9] A. Bruni, S. Mödersheim, F. Nielson, and H. R. Nielson, "Set-pi: Set membership p-calculus," in *IEEE 28th Computer Security Foundations Symposium*, pp. 185–198, 2015.

[10] B. Blanchet, "Automatic verification of security protocols in the symbolic model: The verifier proverif," in *Foundations of Security Analysis and Design VII*, pp. 54–87, 2013.

[11] V. Cortier, F. Eigner, S. Kremer, M. Maffei, and C. Wiedling, "Type-based verification of electronic voting protocols," in *Principles of Security and Trust*, pp. 303–323, 2015.

[12] D. Dolev and A. C. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, 1983.

[13] L. He, Y. Liu, and G. Ren, "Network-layer accountability protocols: A survey," *IEEE Access*, vol. 6, pp. 66886–66902, 2018.

[14] M. S. Hwang, E. F. Cahyadi, Y. C. Chou, and C. Y. Yang, "Cryptanalysis of kumar's remote user authentication scheme with smart card," in *The 14th International Conference on Computational Intelligence and Security (CIS'18)*, pp. 416–420, 2018.

[15] "Implementation of AccountVerif, *Source Code of Protocols and Extended PDF Version.* (`https://www.dropbox.com/sh/tgfim14jvxcrznq/AADefoefTnhlOyKc8b7YGec9a?dl=0`).

[16] B. B. Irawan and M. S. Hwang, "The weakness of moon *et al.*'s password authentication scheme," in *Journal of Physics: Conference Series*, vol. 1069, pp. 012070, 2018.

[17] S. Kremer and R. Künnemann, "Automated analysis of security protocols with global state," in *IEEE Symposium on Security and Privacy*, pp. 163–178, 2014.

[18] R. Künnemann, I. Esiyok, and M. Backes, "Automated verification of accountability in security protocols," in *The 32nd IEEE Computer Security Foundations Symposium (CSF'19)*, pp. 397–413, 2019.

[19] R. Küsters, T. Truderung, and A. Vogt, "Accountability: Definition and relationship to verifiability," in *CCS'10*, pp. 526–535, 2010.

[20] S. Lal and M. L. Das, "On the security analysis of protocols using action language," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 1–9, 2015.

[21] J. Li, H. Lu, and M. Guizani, "ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for vanets," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 938–948, 2015.

[22] E. S. Lowry and C. W. Medlock, "Object code optimization," *Communications of the ACM*, vol. 12, no. 1, pp. 13–22, 1969.

[23] S. Mödersheim, "Abstraction by set-membership: Verifying security protocols and web services with databases," in *CCS*, pp. 351–360, 2010.

[24] B. Schmidt, S. Meier, C. J. F. Cremers, and D. A. Basin, "Automated analysis of diffie-hellman protocols and advanced security properties," in *CSF*, pp. 78–94, 2012.

[25] B. Smyth, M. Ryan, S. Kremer, and M. Kourjieh, "Towards automatic analysis of election verifiability properties," in *ARSPA-WITS*, vol. 6186, pp. 146–163, 2010.

[26] The AVISPA Project, *Avispa Project Deliverable 2.3: The Intermediate Format*, 2003. (`http://www.avispa-project.org/delivs/2.3/d2-3.pdf`)

[27] C. Wang, Y. Xiong, W. Cheng, W. Huang, H. Xia, and J. Huang, "A general formal framework of analyzing selective disclosure attribute-based credential

systems," *International Journal of Network Security*, vol. 19, no. 5, pp. 794–803, 2017.

[28] J. Zhou and D. Gollmann, "A fair non-repudiation protocol," in *IEEE Symposium on Security and Privacy*, pp. 55–61, 1996.

[29] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 429–445, 2019.

Biography

Cheng Su is a Ph.D. candidate in school of Computer Science and Technology, University of Science and Technology of China. His current research interests formal methods and information security.

Wenchao Huang received the B.S. and Ph.D degrees in computer science from University of Science and Technology of China in 2005 and 2011, respectively. He is currently an associate professor in School of Computer Science and Technology, University of Science and Technology of China. His current research interests include mobile computing, information security, trusted computing and formal methods.

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

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

A Practical Method to Attack Deep Learning Based Host Intrusion Detection Systems

Sicong Zhang[1,2], Xiaoyao Xie[1], and Yang Xu[1]
(Corresponding author: Xiaoyao Xie)
Key Laboratory of Information and Computing Science Guizhou Province, Guizhou Normal University[1]
Yunyan District, Guiyang 550001, China
School of Computer Science and Technology, Guizhou University[2]
Huaxi District, Guiyang 550025, China
Email: xyx@gznu.edu.cn
(Received Mar. 29, 2020; Revised and Accepted Dec. 10, 2020; First Online June 23, 2021)

Abstract

The existing adversarial attack methods are not entirely fit for the adversarial tasks in host intrusion detection. Therefore, we propose a new attack method to evaluate better the robustness of deep learning-based host intrusion detection systems against adversarial examples. The proposed method overcomes discreteness in host intrusion detection by perturbing the original inputs with integers. The preliminary results show our method misleads the target classifiers with a high success rate and outperforms the existing state-of-the-art adversarial attack methods. Besides, our method can effectively evade the detection of the classifiers reinforced by state-of-the-art defensive mechanisms.

Keywords: Adversarial Examples; Adversarial Machine Learning; Deep Learning; Intrusion Detection; Neural Networks

1 Introduction

With the success in computer vision [28] and Go [24], deep learning now is widely used in many fields including multi-robot systems [12, 25], health care [23], face recognition [29], and speech recognition [3]. In the domain of cybersecurity, deep learning has also demonstrated its effectiveness, especially in intrusion detection [31] and malware detection [10, 33]. Intrusion detection systems are regarded as one of the most important tools to protect the networks and computer systems. The deep learning techniques are utilized to strengthen the intrusion detection systems because deep learning can extract useful feature representations from the raw data [15].

Though deep learning achieves prominent results [6, 15, 31, 33] in the security area, recent works show its vulnerability to adversarial examples [5, 13, 16, 20, 21, 27], which are derived from adding small but intentionally crafted perturbations to original inputs. Adversarial examples can make deep learning based classifiers misclassify. It is first revealed by Szegedy *et al.* [27] that deep neural networks are vulnerable to adversarially crafted inputs. In computer vision, the perturbations added to original inputs to generate adversarial examples are required to be imperceptible to human eyes, as shown in Figure 1, which comes from [4]. The images in the first row are original, and in the second row are corresponding adversarial examples. Taking the third column as an example, the classification result of GoogLeNet [26] for the original image is bell pepper with 99.9% confidence. However, after adding adversarial perturbations to the original image, the classification result becomes strainer with 86.5% confidence. The two images are almost the same for human eyes. Though adversarial examples are originally proposed in the field of computer vision, previous work has shown that the existence of adversarial examples exposes the weakness of deep learning algorithms and has nothing to do with application scenarios [4]. In cybersecurity, with the extensive application of deep learning, the security of deep learning algorithms has become more and more important. The generation of adversarial examples in cybersecurity has attracted many researchers' interests. Grosse *et al.* [14] show that adversarial examples can make deep neural networks for malware detection misclassify malicious applications as normal with the functionality of malware unchanged. Li *et al.* [17] claim that the adversarial examples produced by their method can evade the detection of the Android malware detection systems protected by firewalls. Yang *et al.* [32] have studied adversarial examples against deep learning based network intrusion detection systems (NIDS). Wang [30] investigates the performance of state-of-the-art adversarial attack methods against deep learning based intrusion detection systems.

The intrusion detection system is a critical tool to protect network and computer systems. The application of

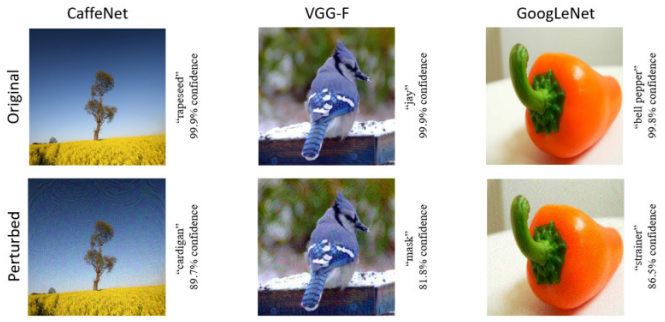


Figure 1: Example of adversarial examples in computer vision

deep learning in intrusion detection is growing rapidly. As a security-critical area, it is very necessary to evaluate the robustness of deep learning based intrusion detection systems against adversarial attacks before their deployment. The common attack methods of generating adversarial examples, such as the box-constrained limited memory approximation of Broyden-Fletcher-Goldfarb-Shanno (LBFGS) [27], fast gradient sign method (FGSM) [13], basic iterative method (BIM) [16], Jacobian-based saliency map attack (JSMA) [21], DeepFool [20], Carlini and Wagner (CW) *et al.* [5] are originally proposed in computer vision in which input domains are continuous, so these algorithms are not entirely fit for the adversarial tasks in cybersecurity in which input domains are usually discrete. Besides, previous studies on adversarial examples in cybersecurity mainly focused on malware detection [7, 14, 17, 19] and network intrusion detection [8, 18, 30, 32]. Host intrusion detection systems (HIDS) [6, 34] are another kind of widely used intrusion detection systems, which detect intrusion behaviors based on system call traces in a target host. There is still a lack of research on adversarial examples against deep learning based host intrusion detection systems (DL-HIDS). Therefore, it is necessary to research the generation of adversarial examples against DL-HIDS to evaluate their robustness.

In this paper, we propose a new attack method called the iterative step method (ISM) to generate adversarial examples against DL-HIDS. It is a white-box and targeted generating algorithm, which requires to access the parameters of the target classifiers and makes the target classifiers output the labels that the adversary specifies. In intrusion detection, this usually means the adversary makes malicious behaviors misclassified as normal behaviors. We train several deep neural networks as the target classifiers on the Australian Defence Force Academy Linux Dataset (ADFA-LD), which achieve state-of-the-art detection performance. We compare the success rate (SR) of ISM against these target classifiers with those of FGSM, BIM, and the Grosse method [14]. We also evaluate the defense performance of some state-of-the-art defensive mechanisms against ISM. The influences of architectures of the classifiers and feature extraction methods on ISM are also investigated. In summary, we make the following contributions:

- To the best of our knowledge, we are the first to investigate the robustness of deep learning based host intrusion detection systems against adversarial examples. The inputs for HIDS are system call traces, which are different from NIDS. We adopt two different kinds of feature extraction methods, *i.e.*, set of words (SOW) and bag of words (BOW), to preprocess the original system call traces. We evaluate the impact of different feature extraction methods on adversarial attacks.
- We propose a new adversarial attack method called the iterative step method to generate adversarial examples against DL-HIDS. The method uses gradients of the loss function to generate adversarial examples discretely. Because of the similarities of the tasks in cybersecurity, we believe our method can be easily extended to evaluate the robustness of other deep learning based systems in cybersecurity such as malware detection systems.
- We train three heterogeneous deep feed-forward neural networks as our DL-HIDS on the ADFA-LD dataset. The impact of different classifiers' architectures on ISM is explored through this.
- Adversarial training [13, 27] and defensive distillation [22] are thought to be the most effective defense methods against adversarial examples. We investigate the defense effect of the two defense methods on ISM.

2 Background

In this section, we provide the background knowledge about this paper. Firstly, we briefly describe HIDS and the feature extraction methods, *i.e.*, SOW and BOW which we adopt to preprocess the original inputs for HIDS. Secondly, we introduce the general concept of deep neural networks which are used to build up our target DL-HIDS. Thirdly, we review several major adversarial attack methods and defense methods. At last, we introduce the ADFA-LD dataset briefly.

2.1 Host Intrusion Detection Systems

Host intrusion detection systems detect abnormal behaviors in a host [2, 6, 34]. That is different from network intrusion detection systems which detect abnormal behaviors in network traffic. HIDS collects the information specific to the operating system of a target host to detect attacks against the target system. In HIDS, system call traces are usually used to describe the behavior in a computer system.

As mentioned above, the original inputs for HIDS are system call traces. They need to be preprocessed before

feeding them to DL-HIDS because DL-HIDS can only process feature vectors. The set of words and bag of words are the most frequently used feature extraction methods of transforming system call traces into feature vectors [2,34] in the domain of host intrusion detection.

Let $\boldsymbol{C} = \{c_1, c_2, c_3, \ldots, c_m\}$ be the set of system calls where m is the number of system calls and c_i denotes a single system call. For any system call trace $\boldsymbol{s}$, SOW transforms it into a vector $\boldsymbol{x} =< x_1, x_2, x_3, \ldots, x_m >$, where $x_i = 1$ if $\boldsymbol{s}$ includes c_i else $x_i = 0$. BOW does the transformation in much the same way except x_i represents the number of occurrences of c_i in $\boldsymbol{s}$.

2.2 Deep Neural Networks

Neural networks have many variants including feed-forward neural networks (FNN), convolutional neural networks (CNN), recurrent neural networks (RNN), *et al.* In general, neural networks consist of the input layer, the hidden layer, and the output layer. When the number of hidden layers exceeds two, neural networks are regarded as deep neural networks (DNN). We choose deep feed-forward neural networks (DFNN) as our target DL-HIDS because DFNN is widely used in cybersecurity [14,15,30,32]. The common architecture of DFNN is shown in Figure 2.

The basic computing units in DFNN are called neurons. Each layer in DFNN consists of several neurons, through which the previous layers are connected to the next layers. The output of the previous layer is the input of the next layer. Activation functions are used to improve the nonlinearity of DFNN. The DFNN can be formalized as Equation (1) where $F(\boldsymbol{x})$ denotes the output of DFNN and f_i denotes the output of the ith layer, $i = 1, 2, \ldots, l$. l is the number of layers in DFNN except the input layer. The output layer of DFNN is usually a softmax layer in which the number of neurons is the same as the number of classes. Let $F_i(\boldsymbol{x})$ is the output of ith neuron in the output layer where $i = 1, 2, 3, \ldots, n$ and n is the number of classes. It denotes the probability of input $\boldsymbol{x}$ belonging to class i. Therefore, $0 \leq F_i(\boldsymbol{x}) \leq 1$ and $\sum_i F_i(\boldsymbol{x}) = 1$.

$$F(\boldsymbol{x}) = f_l(\ldots f_2(f_1(\boldsymbol{x}))). \tag{1}$$

2.3 Adversarial Attack and Defense Methods

2.3.1 Attack Methods

The goal of adversarial attacks is to modify the original inputs as little as possible to make the target classifiers output the wrong results in computer vision [4]. Attack methods of generating adversarial examples can be categorized as white-box and black-box according to the knowledge of the target classifiers possessed by the adversary. In the context of white-box, the adversary possesses the complete information of the target classifiers including architectures and parameters. The black-box attacks are performed when adversaries possess limited knowledge of the target classifiers which excludes the model parameters [4]. The white-box attacks can also be used to evaluate the target classifiers in a black-box way because of the transferability of the adversarial examples, which implies that adversarial examples generated on a target model can be utilized to attack another model with a different architecture [11]. Attack methods can also be categorized as targeted and nontargeted according to the adversarial goals. Given a classifier $F(\boldsymbol{x})$, let $C(\boldsymbol{x}) = argmax_i F_i(\boldsymbol{x})$ be the label predicted by the classifier, $\boldsymbol{x}$ be the original input, $\boldsymbol{x}^*$ be the corresponding adversarial example, y be the real label of $\boldsymbol{x}$, and y^* be the target label which is the label that adversaries want the classifier to output. The goal of nontargeted attacks is to find a $\boldsymbol{x}^*$ which makes the classifier output $C(\boldsymbol{x}^*) \neq y$. The goal of targeted attacks is to make the classifier output $C(\boldsymbol{x}^*) = y^*$.

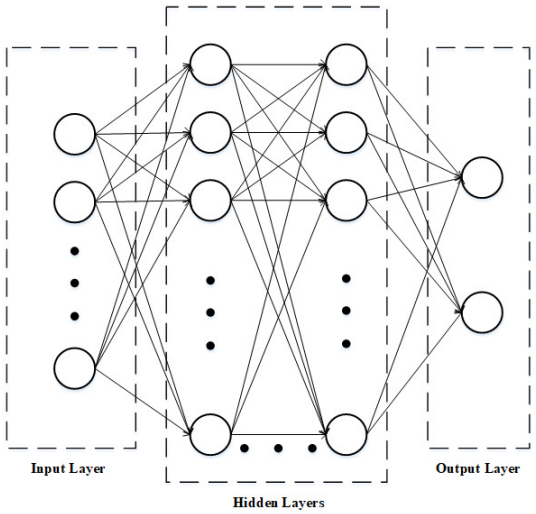


Figure 2: The architecture of deep feed-forward neural networks

In the remainder of this section, we introduce several widely used attack methods which we compare with our method in Section 4. All methods introduced here are their non-targeted version.

1) Fast gradient sign method (FGSM).

   FGSM [13] is a one-step and white-box method, which generates adversarial examples by maximizing the loss function $J(\boldsymbol{x}, y)$. It can be formalized as Equation (2), where $\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y)$ is the gradient of the loss function with respect to $\boldsymbol{x}$. θ limits the perturbations to meet the bound $||\boldsymbol{x}^* - \boldsymbol{x}||_\infty \leq \theta$. The *sign* function carry out the following mapping: $\boldsymbol{x} < 0$, $sign(\boldsymbol{x}) = -1$; $\boldsymbol{x} = 0$, $sign(\boldsymbol{x}) = 0$; $\boldsymbol{x} > 0$, $sign(\boldsymbol{x}) = 1$.

$$\boldsymbol{x}^* = \boldsymbol{x} + \theta \times sign(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y)). \tag{2}$$

2) Basic iterative method (BIM).

   BIM [16] is an iterative version of FGSM, which can be formalized as Equation (3), where α is the same

as θ but with a smaller value. $\boldsymbol{x}_i^*$ is the adversarial example produced during the ith iteration, where $i = 0, 1, \ldots, t-1$. t is the maximum number of iterations and $\boldsymbol{x}_0^* = \boldsymbol{x}$. Iterative methods are believed to be stronger than one-step ones but with worse transferability [11].

$$\boldsymbol{x}_{i+1}^* = \boldsymbol{x}_{\boldsymbol{i}}^* + \alpha \times sign(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_i^*, y)). \quad (3)$$

3) Grosse method (GM).

Unlike the two methods above, which are proposed to generate adversarial examples against the classifiers in the field of computer vision, Grosse *et al.* [14] expand the JSMA [21] to generate adversarial examples against deep learning based malware classifiers. GM perturbs the input features based on the Jacobian matrix of $F(\boldsymbol{x})$ with respect to $\boldsymbol{x}$. In each iteration, GM set $x_i = 1$ where x_i is the ith component of $\boldsymbol{x}$ and i is given by Equation (4). m is the number of components in $\boldsymbol{x}$.

$$i = argmax_{j \in [1,m], x_j = 0} \frac{\partial F_{y^*}(\boldsymbol{x})}{\partial \boldsymbol{x}_j}. \quad (4)$$

2.3.2 Defense Methods

The study of adversarial attacks and defenses is like a competition. When some new attack methods come out and perform well, the corresponding defense methods will be proposed soon. Since the emergence of adversarial examples, a lot of defense methods have been proposed to resist the adversarial attacks [4]. Among all the defense methods, adversarial training and defensive distillation are considered to be two of the most effective defense methods. In this section, we briefly introduce these two methods, which are used to validate the effectiveness of ISM.

1) Adversarial training.

Adversarial training is thought to be one of the most effective defensive mechanisms [4, 13, 27] against adversarial examples. Previous work chooses adversarial training as the defense method to verify their newly proposed attack methods.

The main idea of adversarial training is to train the classifiers with additional adversarial examples generated by corresponding attack methods. The procedure of adversarial training is shown in Figure 3. Adversarial training can not only improve the robustness of the classifiers against adversarial examples but also improve the model's generalization. Adversarial examples used in Step 3 are generated in Step 2 and labeled as their original classes.

2) Defensive distillation.

The distillation technique is firstly proposed to transfer the knowledge of a neural network to another one. Papernot *et al.* [22] extend the distillation technique to defend the adversarial attacks. The procedure of defensive distillation is shown in Figure 4.

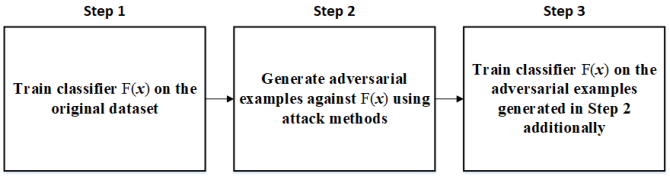


Figure 3: The procedure of adversarial training

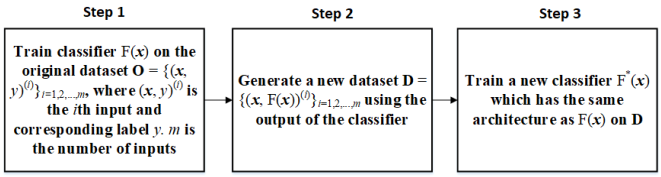


Figure 4: The procedure of defensive distillation

The output of the target classifier is not a normal softmax layer in distillation. The change for $F(\boldsymbol{x})$ is shown in Equation (5) where $i = 1, 2, \ldots, n$ and n is the number of classes. The $Z_i(\boldsymbol{x})$ is the ith component of the output of the classifier before the softmax layer. T is the distillation temperature, which controls the effect of defensive distillation. When T is set to 1, it goes back to a normal softmax layer. Normally, T is set to a high value to obtain a good distillation result [22].

$$F_i(\boldsymbol{x}) = \frac{e^{Z_i(\boldsymbol{x})/T}}{\sum_{l=1}^{n} e^{Z_l(\boldsymbol{x})/T}}. \quad (5)$$

2.4 ADFA-LD Dataset

ADFA-LD dataset is published by Creech *et al.* [9]. We choose it as our experimental dataset because it is the latest professional dataset for HIDS. ADFA-LD dataset contains 833 normal training traces, 4372 normal validation traces, and 746 attack traces. All the traces are collected under the Linux operating system. Attack traces include six major and common attacks. The details of the ADFA-LD dataset are shown in Table 1.

Table 1: Details of the ADFA-LD dataset

Data Type		Traces
Normal Data	Training Data	833
	Validation Data	4372
Attack Data	Adduser	91
	Hydra_FTP	162
	Hydra_SSH	176
	Java_Meterpreter	124
	Meterpreter	75
	Web Shell	118

3 Methodology

The existing adversarial attack methods including FGSM, BIM, JSMA, *et al.* are originally proposed to generate adversarial examples against classifiers in computer vision. These methods are not entirely fit for adversarial tasks in the domain of cybersecurity because there are two obvious differences between the two fields [4, 14]:

- The input domains in cybersecurity are usually discrete. In contrast, the input domains in computer vision are continuous. This means the perturbations crafted by the attack methods are no longer small values but integers. For example, the input domain for HIDS is sometimes binary. The modification to these features can only be $0 \rightarrow 1$.
- In computer vision, the perturbations added to original inputs are required to be imperceptible to human eyes. In cybersecurity, the restriction on imperceptibility is meaningless. It is replaced by guaranteeing the validity of the input. In other words, after modifying the intrusion data to evade the detection of DL-HIDS, the intrusion data can still achieve its original intrusion purpose.

Although we can simply expand these existing methods in the same discrete way described in Algorithm 1 to make them suitable for generating adversarial examples against DL-HIDS, they do not work well. We verify this in Section 4 later. Currently, the researches on adversarial examples mainly focus on computer vision. More adversarial attack methods against deep learning based systems in cybersecurity need to be proposed to evaluate the robustness of these systems.

In this paper, to better evaluate the robustness of DL-HIDS against adversarial examples, we propose a new method called the iterative step method to generate adversarial examples against DL-HIDS. Although this method is proposed to evaluate the robustness of DL-HIDS, it can be easily extended to other tasks in cybersecurity such as malware detection because of the similar discreteness of these tasks [14, 32]. The restrictions on validity are also fit for the other tasks in this domain.

In this section, we introduce how to guarantee the validity of inputs when crafting adversarial examples against DL-HIDS and we discuss the details of ISM.

3.1 Validity of Inputs

It is necessary to guarantee the validity of adversarial examples against DL-HIDS. If the purpose of a system call trace of a malicious process is to obtain administrators' passwords, it is meaningless that after disguising the original malicious trace as normal, it can evade the detection of DL-HIDS but lose the ability to obtain the passwords. No attackers are willing to do such things.

Inputs for DL-HIDS are system call traces which are usually represented as feature vectors using the feature extraction methods discussed in Section 2. Each feature in a vector represents the occurrence or the number of occurrences of a system call. Therefore, if we just add system calls to system call traces and never remove them, then we will not influence its original functionality. Besides, we can limit the number and types of system calls which can be modified to further ensure the validity of the adversarially crafted traces [14, 30].

In the real world, it is feasible for the adversary to increase the number of some system calls in a specific process. Therefore, we believe our method of guaranteeing the validity of inputs is practical.

3.2 Iterative Step Method

Generating an adversarial example in a targeted way can be formalized as Equation (6) where $\boldsymbol{x}$ is the original input, y^* is the target label, and $\boldsymbol{\delta}$ is the minimal adversarial perturbation causing $F(\boldsymbol{x})$ to misclassify. Normally, $\boldsymbol{\delta}$ may not be unique.

$$\begin{aligned} &\text{minimize} \quad ||\boldsymbol{\delta}||_p \\ &\text{subject to} \quad C(\boldsymbol{x}+\boldsymbol{\delta}) = y^*. \end{aligned} \tag{6}$$

The existing attack methods such as FGSM and BIM generate adversarial examples by minimizing the loss function $J_F(\boldsymbol{x}, y^*)$. The loss function is usually the cross-entropy loss. These methods use the gradient of the loss function to construct the adversarial perturbation and modify the original input with all the components of the perturbation vector to generate adversarial examples. This is possible in computer vision because every pixel in an image can be modified as long as the corresponding perturbation component is small enough. However, in host intrusion detection, not every component of the input vector can be perturbed because of the restriction on the validity of inputs. This limits the direct application of these methods in host intrusion detection. The GM uses the Jacobian matrix to decide the component to be perturbed in each iteration. Inspired by that, we choose the components to be perturbed by simply sorting the components of the gradient of the loss function in each iteration. Our method can be regarded as an improvement and synthesis of these two kinds of methods. The generating process of ISM can be formalized as Equation (7) where $\boldsymbol{x}_i^*$ is the adversarial example produced during the ith iteration, $i = 0, 1, \ldots, t-1$, t is the maximum number of iterations, and $\boldsymbol{x}_0^* = \boldsymbol{x}$. Our method also generates the adversarial perturbation by computing the gradient of $J_F(\boldsymbol{x}, y^*)$ with respect to $\boldsymbol{x}$ and obtains the adversarial example by subtracting the perturbation from the original input iteratively. However, our method does not simply modify the input with all the components of the perturbation vector in each iteration just like FGSM and BIM. We sort out the features of $\boldsymbol{x}_i^*$ that can be perturbed in each iteration and make sure the perturbation generated in each iteration are integers. Then, the first few components of the filtered perturbation will be used to perturb

the input. The intact pseudocode of the iterative step method is shown in Algorithm 1.

$$\boldsymbol{x}^*_{i+1} = \boldsymbol{x}^*_i - \epsilon \times sign(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}^*_i, y^*)). \quad (7)$$

ISM is not just fit for DFNN but can be extended to any differentiable classifiers. In Algorithm 1, for each iteration, the method firstly computes the gradient $\boldsymbol{g}$ for the current value of $\boldsymbol{x}$ in Step 3. Secondly, it sorts $\boldsymbol{g}$ in ascending order. To guarantee the validity of inputs, we only add additional system calls into original inputs. Therefore, we only select components of $\boldsymbol{x}$ whose gradients are less than or equal to 0 in Step 6. We further pick out the components in $\boldsymbol{x}$ which are equal to 0 in Step 7 from the result of the previous step. This is because the input vector for DL-HIDS is binary in the setting of SOW. We can only perturb the components which are zero. In the setting of BOW or other nonbinary cases, this restriction can be removed. At last, to restrict the modifications to $\boldsymbol{x}$, the first k components of filtered perturbations are utilized to perturb $\boldsymbol{x}$. Although we do not need to modify the inputs as little as possible in host intrusion detection just like in computer vision, it is suggested to restrict the modification to the inputs because less modification further guarantees the validity of the inputs. The *sign* function ensures the perturbations added to the original inputs to be integers. k and ϵ can be just set to integers greater than zero to further ensure the perturbations to be integers. The modified features always change from one state to another just like the step function. This is why we call it the iterative step method.

Three conditions terminate the while loop:

1) The target classifiers output the target label;
2) The maximum number of iterations is reached;
3) In an iteration, the satisfying components are not found.

The number of perturbed features k and perturbation strength ϵ are two important parameters for ISM. They control the distances between original inputs and adversarial examples. They also influence the attack effect of ISM. The distances measure the differences between original inputs and adversarial examples. Previous work employs l_p norms to measure the distances. In the context of DL-HIDS, we believe l_1 norm is the best way to describe the differences between original inputs and adversarial examples. It intuitively denotes the number of system calls added to original inputs. The details of how k and ϵ influence the distances and attack effect are discussed in Section 4.

Algorithm 1 Iterative Step Method

Input: Classifer F; loss function J; the original input $\boldsymbol{x}$; the target label y^*;
Max iterations m; the number of perturbed features k;
Perturbation strength ϵ.
Output: adversarial examples **adx**

1: $j \leftarrow 0$
2: **while** $\text{argmax}_i \text{F}_i(\boldsymbol{x}) != y^*$ **and** $j < m$ **do**
3: Compute the gradient $\nabla_{\boldsymbol{x}} \text{J}_\text{F}(\boldsymbol{x}, y^*)$
4: $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{x}} \text{J}_\text{F}(\boldsymbol{x}, y^*)$
5: Sort $\boldsymbol{g}$ in ascending order and return the sorted index of $\boldsymbol{g}$:
Ind $\leftarrow$ argsort($\boldsymbol{g}$)
6: select componets of $\boldsymbol{x}$ whose gradients are less than or equal to 0:
Ind $\leftarrow$ **Ind**[$\boldsymbol{g}$[**Ind**] <= 0]
7: select componets of $\boldsymbol{x}$ which are equal to 0:
Ind $\leftarrow$ **Ind**[$\boldsymbol{x}$[**Ind**] == 0]
8: **if** len(**Ind**) $< k$ **then**
9: break
10: **end if**
11: Update $\boldsymbol{x}$ with filtered gradients
$\boldsymbol{x}$[**Ind**[: k]] $\leftarrow$ $\boldsymbol{x}$[**Ind**[: k]] $- \epsilon \cdot$ sign($\boldsymbol{g}$[**Ind**[: k]])
12: $j \leftarrow j + 1$
13: **end while**
14: **adx** $\leftarrow \boldsymbol{x}$
15: **return adx**

4 Experiments and Analysis

In this section, we train three DFNNs, which have different architectures, as our target DL-HIDSs that achieve the detection performance comparable to state-of-the-art HIDS in the literature [2, 6, 9, 34]. These classifiers are trained on the ADFA-LD dataset and used to validate the effectiveness of ISM. We explore the influences of architectures of the target classifiers on ISM through these heterogeneous classifiers. The original system call traces for DL-HIDS are preprocessed by two different feature extraction methods, *i.e.*, SOW and BOW as discussed in Section 2. Therefore, the impact of different feature extraction methods on ISM is investigated, too. At the end of this section, we investigate the effect of defense methods on ISM, *i.e.*, adversarial training and defensive distillation.

Intrusion detection is usually a binary classification task. Inputs for DL-HIDS are labeled as normal (0) or attack (1). The metrics used in our experiments include accuracy (ACC), precision (PR), recall (RC), false alarm (FA), success rate (SR), as shown in Equations (8)-(12). The formulas (8)-(11) are used to evaluate the performance of the target classifiers. The formula (12) is used to evaluate the attack effect of ISM against the target classifiers.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$PR = \frac{TP}{TP + FP} \quad (9)$$

$$RC = \frac{TP}{TP + FN} \quad (10)$$

$$FA = \frac{FP}{TN + FP} \quad (11)$$

$$SR = \frac{AE}{OI} \quad (12)$$

The definitions for the terms in these formulas are as below:

1) True Positive (TP): Attack samples are correctly classified as attack.
2) True Negative (TN): Normal samples are correctly classified as normal.
3) False Positive (FP): Normal samples are incorrectly classified as attack.
4) False Negative (FN): Attack samples are incorrectly classified as normal.
5) Adversarial Examples (AE): The number of adversarial examples that succeed to make the target classifiers misclassify.
6) Original Inputs (OI): The number of correctly classified original inputs that are used to generate adversarial examples.
7) Accuracy: The percentage of the correctly classified samples over the total number of samples.
8) Precision: The percentage of the correctly classified attack samples over the total number of samples which are predicted to be attack.
9) Recall: The percentage of the correctly classified attack samples over the total number of attack samples.
10) False Alarm: The percentage of the incorrectly classified normal samples over the total number of normal samples.
11) Success Rate: The percentage of successful attacks against the target systems over the total number of attacks.

We implement all the experimental code based on Tensorflow-GPU 1.6.0 [1] which is the most widely used deep learning framework and a 3GB Nvidia GTX 1060 GPU is used to accelerate the computing.

4.1 Training Target Classifiers

We train three DFNNs as our target DL-HIDS on the ADFA-LD dataset. All the DFNNs are binary classifiers that classify inputs as normal or attack. The three target classifiers, which achieve the detection performance comparable to state-of-the-art HIDS [2,6,9,34], have different architectures as shown in Table 2. We adopt different feature extraction methods to preprocess the original system call traces, *i.e.*, SOW and BOW. Therefore, every classifier has two versions. After the preprocessing, every system call trace is transformed into an input vector with 175 features.

[500,500,500] denotes that the classifier has three hidden layers and each layer has 500 neurons, and so on in the other cases. The classifier C1 has the same number of hidden layers as that of the classifier C2 but the different number of neurons in each hidden layer. The classifier C3 has the same number of neurons as that of the classifier C2 in each hidden layer but the different number of hidden layers. We investigate the influence of the different architectures on ISM through these heterogeneous classifiers. Different versions of the same classifier are used to explore the impact of different feature extraction methods on ISM.

The 30 percent of the ADFA-LD dataset is used as the testing set and the rest is used as the training set. The batch size is 256 and the epoch is set to 50. Adam is adopted as the optimizer to train all the classifiers. We adopt the same parameters to train all the target classifiers to avoid the influences of other factors on the experiments. The details of hyper-parameters used to train the target classifiers are shown in Table 3 where LR is short for learning rate and β_1, β_2, and ϵ are the hyper-parameters of Adam optimizer.

4.2 Crafting Adversarial Examples Against DL-HIDS

The purpose of adversarial attacks is to mislead the classifiers. There are two major objectives in adversarial deep learning, *i.e.*, integrity and availability violations [8]. In the context of host intrusion detection, integrity violations attempt to mislead the classifiers to classify the attack data as normal and availability violations attempt to make the normal data classified as attack. We assume that the adversarial attacks against DL-HIDS happen in the test phase, which is closer to the actual condition. Therefore, we use the test set to craft the adversarial examples.

In this section, we adopt ISM to generate adversarial examples against DFNNs for host intrusion detection which are trained in Section 4.1. We evaluate the effect of ISM on integrity and availability violations. As mentioned in Section 3.2, the number of perturbed features k and perturbation strength ϵ are two very important parameters that control the SR of ISM and distances between original inputs and adversarial examples. We explore the influence of these two parameters in the experiments.

We adopt l_p norms to measure the distances between original inputs and corresponding adversarial examples. The smaller the distances are, the fewer modifications we make to original inputs. We adopt common l_0, l_1, l_2, and l_∞ norms as our distance metrics. The general formula for computing l_p norm is shown in Equation (13) where $\boldsymbol{d}$ denotes the differences between original inputs and corresponding adversarial examples. m denotes the number of components in $\boldsymbol{d}$. d_i denotes the ith component of $\boldsymbol{d}$. Normally, l_0 norm measures the number of non-zero components in a vector. l_1 norm indicates the sum of the absolute values of components in a vector. l_2 norm measures the classical euclidean distance. l_∞ norm measures the maximum of the absolute values of components in a

Table 2: Target classifiers for host intrusion detection

Classifiers	Architecture	ACC(%)	PR(%)	RC(%)	FA(%)
C1_SOW	[500,500,500]	95.13	79.83	83.62	3.15
C1_BOW	[500,500,500]	96.41	83.39	90.56	2.70
C2_SOW	[1000,1000,1000]	95.02	80.17	81.90	3.02
C2_BOW	[1000,1000,1000]	96.70	86.55	88.41	2.06
C3_SOW	[1000,1000,1000,1000]	95.35	81.17	83.62	2.90
C3_BOW	[1000,1000,1000,1000]	95.58	82.08	84.55	2.77

Table 3: Experimental setup for training the target classifiers

Activation	LR	β_1	β_2	ϵ
ReLU	$1e^{-3}$	0.9	0.999	$1e^{-8}$

vector.

$$l_p = ||\boldsymbol{d}||_p = \sqrt[p]{\sum_{i=1}^{m} d_i^p} \tag{13}$$

4.2.1 Adversarial Examples Against Sow-based Classifiers

SOW transforms original system call traces into binary feature vectors $\boldsymbol{x} \in \{0,1\}^m$ as discussed in Section 2. To guarantee the validity of inputs, ISM only adds system calls to original inputs. Parameters k and ϵ of ISM are used to control the attack effect and distances. In the setting of SOW, we can only set zero components in $\boldsymbol{x}$ to one, as shown in Step 7 of Algorithm 1 and the ϵ can be just set to 1.

k describes the number of features perturbed in each iteration and ϵ indicates the perturbation strength. The product of k and ϵ corresponds to the number of system calls added to inputs in each iteration. In the context of integrity attacks, the influences of k on attack effect and distances are shown in Figure 5. We adopt C1_SOW as the target classifier to obtain the results in Figure 5. The distances in Figure 5 are measured by l_1 norm.

As shown in Figure 5, with the increase of k, the success rate of ISM firstly maintains a high level and distances between original inputs and corresponding adversarial examples are gradually increasing. But when k approaches 90, the success rate and distances begin to decrease. The intuitive hypothesis for this phenomenon is that the number of features which can be used to craft adversarial examples is limited. On the other hand, this means ISM just needs to modify a very small part of features in original inputs to generate the adversarial examples which can mislead the target classifiers. This makes our ISM feasible and practical in the real environment.

We compare the attack effect of ISM with those of GM, FGSM, and BIM. FGSM and BIM are not directly appropriate for generating adversarial examples against DL-HIDS. Therefore, we extend the two methods by adding the same restrictions of Step 6 and Step 7 in Algorithm 1 on them to make them fit for generating adversarial examples against DL-HIDS. The θ for FGSM and α for BIM are set to integers too.

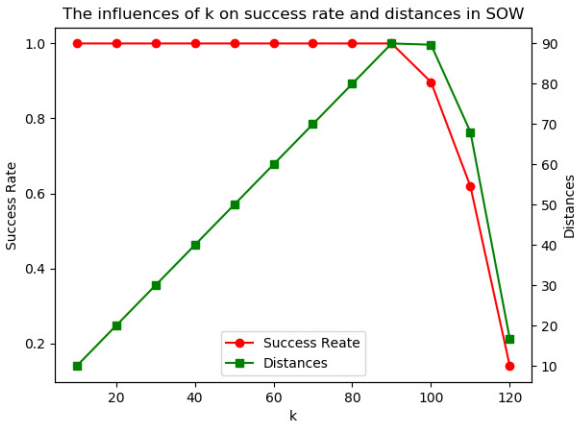


Figure 5: Influences of k on success rate and distances in the setting of SOW (The results in Figure 5 are obtained by fixing ϵ to be 1 and setting the maximum number of iterations m to 50)

The results of integrity attacks are shown in Table 4. EFGSM and EBIM are the extended versions of FGSM and BIM. We set k to 1, ϵ to 1, and the maximum number of iterations to 50 for ISM. We set θ to 1 for EFGSM and α to 1 for EBIM. The maximum number of iterations for EBIM is set to 2. The maximum number of perturbed features for GM is set to 20. We only select the original inputs which are correctly classified to generate the adversarial examples. All the distances in Table 4 are averaged on all the selected original inputs.

The results in Table 4 show that all the methods achieve a high success rate in attacking the SOW-based classifiers. However, ISM and GM add far fewer system calls to original inputs than EFGSM and EBIM. This means ISM and GM are more practical in the field of host intrusion detection. ISM achieves the same performance as GM in the setting of SOW.

Table 5 shows the results of availability attacks against SOW-based classifiers. The same experimental param-

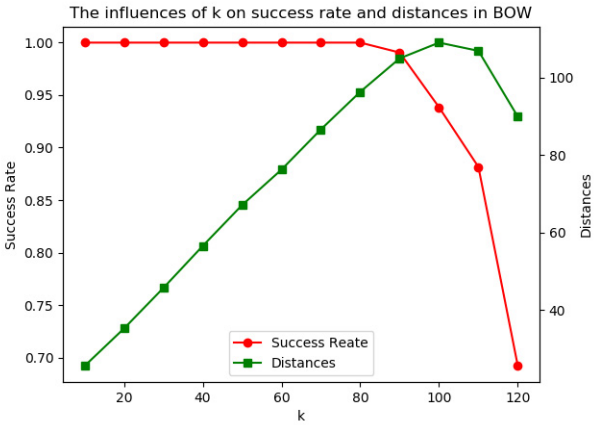


Figure 6: Influences of k on success rate and distances in the setting of BOW (The results in Figure 6 are obtained by fixing ϵ to be 1 and setting the maximum number of iterations m to 50)

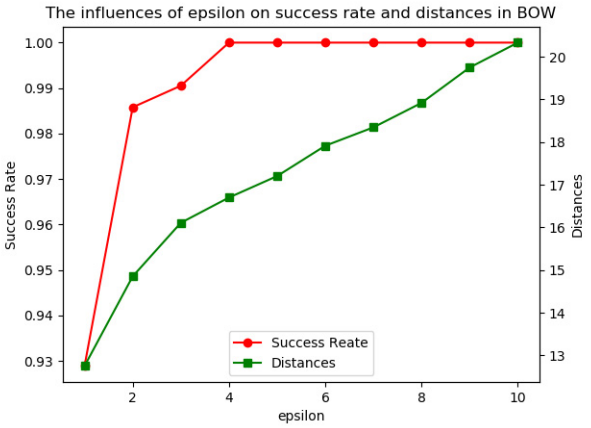


Figure 7: Influences of ϵ on success rate and distances in the setting of BOW (The results in Figure 7 are obtained by fixing k to be 1 and setting the maximum number of iterations m to 50)

eters as those in Table 4 for the attack methods are adopted. From the results in Table 5, we conclude that the success rate of availability attacks from these methods is lower than that of integrity attacks. In contrast, the distances increase.

Comparing the same metrics of different classifiers in the same table, we can conclude that the architectures of target classifiers have a small impact on the attack effect in the setting of SOW. However, the more complex the architectures are, the more modifications the attack methods generally need to make to the original inputs.

4.2.2 Adversarial Examples Against Bow-based Classifiers

BOW transforms the original inputs in much the same way as SOW except that each component in the vectors represents the number of occurrences of a system call, as discussed in Section 2. Therefore, the restriction of Step 7 in Algorithm 1 can be removed. ϵ can be set to any integer greater than 0. We investigate the influences of k and ϵ on the attack effect and distances in the setting of BOW. The influences of the two parameters on integrity attacks are shown in Figure 6 and Figure 7. The C1_BOW is adopted as the target classifier to obtain the results in Figure 6 and Figure 7. The distances are also measured by l_1 norm.

The trend of k in Figure 6 is the same as that in Figure 5. In Figure 7, we can see that with the increase of ϵ, the success rate and distances between original inputs and adversarial examples are increasing. When ϵ is large enough, ISM maintains a high success rate.

The results of integrity and availability attacks against BOW-based classifiers are shown in Table 6 and Table 7. The results in Table 6 and Table 7 are obtained by adopting the same parameters for these attack methods. For ISM, we set both k and ϵ to 2. The maximum number of iterations is still set to 50. We set θ to 3 for EFGSM and α to 2 for EBIM. The maximum number of iterations for EBIM is set to 2. The maximum number of perturbed features for GM is set to 100.

The results show that ISM outperforms the other three methods in the success rate. GM produces the shortest distances. ISM is not far behind it concerning the distances. In the setting of BOW, the architectures of the target classifiers have a bigger impact on the success rate. In general, classifiers that have more layers and neurons are less vulnerable to adversarial examples.

Comparing the results in Table 6 and Table 7 with those in Table 4 and Table 5, we can conclude that BOW-based classifiers are more robust against adversarial examples than SOW-based classifiers.

4.3 Defense Adversarial Attacks

Since the emergence of adversarial examples, a lot of methods are proposed to defend the adversarial attacks. Adversarial training and defensive distillation are thought to be the most effective mechanisms against adversarial attacks. In this section, we adopt the two defensive methods to further evaluate the effectiveness of ISM.

In the real world, the adversary is more likely to disguise malicious behaviors as normal. Therefore, in this section, we only investigate defenses against integrity attacks.

4.3.1 Adversarial Training

Adversarial training is thought to be a brute-force way to reinforce the target model [4] by training the target classifiers with additional adversarial examples. Adversarial training can improve not only the robustness of the target classifiers against adversarial examples but also the model's generalization.

Table 4: Results of integrity attacks against SOW-based classifiers

Classifiers	Attack methods	SR(%)	l_0	l_1	l_2	l_∞
C1_SOW	ISM	100.00	1.70	1.70	1.27	1.00
	GM	100.00	1.69	1.69	1.27	1.00
	EFGSM	100.00	110.82	110.82	10.52	1.00
	EBIM	100.00	110.82	110.82	10.52	1.00
C2_SOW	ISM	100.00	1.59	1.59	1.23	1.00
	GM	100.00	1.59	1.59	1.23	1.00
	EFGSM	100.00	113.16	113.16	10.63	1.00
	EBIM	100.00	113.16	113.16	10.63	1.00
C3_SOW	ISM	100.00	1.93	1.93	1.36	1.00
	GM	100.00	1.93	1.93	1.36	1.00
	EFGSM	100.00	114.96	114.96	10.72	1.00
	EBIM	100.00	114.96	114.96	10.72	1.00

Table 5: Results of availability attacks against SOW-based classifiers

Classifiers	Attack methods	SR(%)	l_0	l_1	l_2	l_∞
C1_SOW	ISM	99.86	2.68	2.68	1.58	1.00
	GM	99.87	2.65	2.65	1.57	1.00
	EFGSM	99.60	46.43	46.43	6.79	1.00
	EBIM	99.73	67.43	67.43	8.19	1.00
C2_SOW	ISM	99.87	2.76	2.76	1.59	1.00
	GM	99.87	2.72	2.72	1.59	1.00
	EFGSM	99.73	47.28	47.28	6.87	1.00
	EBIM	99.87	73.89	73.89	8.58	1.00
C3_SOW	ISM	99.87	3.52	3.52	1.81	1.00
	GM	99.87	3.48	3.48	1.80	1.00
	EFGSM	99.73	48.87	48.87	6.97	1.00
	EBIM	99.87	74.53	74.53	8.62	1.00

Table 6: Results of integrity attacks against BOW-based classifiers

Classifiers	Attack methods	SR(%)	l_0	l_1	l_2	l_∞
C1_BOW	ISM	100.00	2.60	17.69	11.02	8.23
	GM	98.58	1.60	15.15	12.93	12.22
	EFGSM	98.56	120.41	361.23	32.89	3.00
	EBIM	98.58	135.93	392.26	34.15	3.18
C2_BOW	ISM	98.55	2.59	16.17	9.79	7.06
	GM	94.66	1.88	17.61	13.13	11.93
	EFGSM	97.57	127.15	381.43	33.78	3.00
	EBIM	98.55	138.75	378.11	32.41	2.92
C3_BOW	ISM	99.07	2.58	21.08	13.58	10.12
	GM	95.81	1.36	16.63	14.86	14.33
	EFGSM	94.88	118.93	356.78	32.62	3.00
	EBIM	98.55	138.75	378.11	32.41	2.92

Table 7: Results of availability attacks against BOW-based classifiers

Classifiers	Attack methods	SR(%)	l_0	l_1	l_2	l_∞
C1_BOW	ISM	96.69	2.98	39.50	24.51	18.46
	GM	91.00	1.66	31.73	27.72	26.75
	EFGSM	67.37	45.75	137.25	20.27	3.00
	EBIM	77.56	55.86	185.01	25.59	4.00
C2_BOW	ISM	95.79	2.82	44.55	28.79	21.82
	GM	90.13	1.41	33.75	31.28	30.67
	EFGSM	59.70	41.87	125.62	19.39	3.00
	EBIM	70.41	49.58	167.97	24.69	4.00
C3_BOW	ISM	94.65	2.53	43.92	28.81	21.22
	GM	89.89	1.51	33.76	30.13	29.18
	EFGSM	62.22	39.75	119.25	18.87	3.00
	EBIM	70.61	49.77	163.99	24.17	4.00

The procedure of adversarial training is discussed in Section 2. We adopt Equation 14 as the new loss function to retrain our target classifiers [13]. $\boldsymbol{x}^*$ is the adversarial examples crafted from original input $\boldsymbol{x}$. We set σ to 0.5. The results of adversarial training against ISM are shown in Table 8.

We adopt the same parameters as those in Section 4.2 for ISM. Classifiers with the suffix AT in Table 8 are adversarially trained ones. Classifiers without the suffix AT are the normal ones introduced in Table 2.

Results in Table 8 show that the combination of BOW and adversarial training decreases the success rate of ISM and increases the distances used to generate adversarial examples. The increase of distances means adversary needs to make more modifications to original inputs to generate adversarial examples that can mislead the target classifiers. ISM shows a good attack effect on adversarially trained SOW-based classifiers.

$$\widetilde{J}(\boldsymbol{x}, y) = \sigma \cdot J(\boldsymbol{x}, y) + (1 - \sigma) \cdot J(\boldsymbol{x}^*, y). \quad (14)$$

Table 8: Results of adversarial training against integrity attacks from ISM

Classifiers	SR(%)	l_0	l_1	l_2	l_∞
C1_SOW_AT	100.00	2.36	2.36	1.46	1.00
C1_SOW	100.00	1.70	1.70	1.27	1.00
C1_BOW_AT	94.65	3.71	48.30	28.83	21.95
C1_BOW	100.00	2.60	17.69	11.02	8.23
C2_SOW_AT	100.00	2.37	2.37	1.49	1.00
C2_SOW	100.00	1.59	1.59	1.23	1.00
C2_BOW_AT	94.85	3.56	48.88	28.82	21.56
C2_BOW	98.55	2.59	16.17	9.79	7.06
C3_SOW_AT	100.00	2.31	2.31	1.46	1.00
C3_SOW	100.00	1.93	1.93	1.36	1.00
C3_BOW_AT	91.53	3.86	61.06	36.39	27.56
C3_BOW	99.07	2.58	21.08	13.58	10.12

4.3.2 Defensive Distillation

The procedure of defensive distillation is discussed in Section 2. The parameter T of defensive distillation is set to 20. The results are shown in Table 9.

The same parameters as those in Section 4.2 for ISM are adopted. Classifiers with the suffix DD in Table 9 are the ones reinforced by defensive distillation.

Just like the results in Table 8, the combination of BOW and defensive distillation decreases the success rate of ISM and increases the distances used to generate adversarial examples. However, the defense performance of adversarial training against ISM is better than that of defensive distillation. The SOW-based classifiers are more vulnerable to adversarial examples crafted by ISM. We observe an intriguing phenomenon from the results in Table 9 that the combination of SOW and defensive distillation does not decrease the success rate of ISM but decreases the distances needed to generate the adversarial examples. We don't know how and why this happens and more experiments on different datasets are needed to verify this result. We leave it to our future work.

Table 9: Results of defensive distillation against integrity attacks from ISM

Classifiers	SR(%)	l_0	l_1	l_2	l_∞
C1_SOW_DD	100.00	1.42	1.42	1.17	1.00
C1_SOW	100.00	1.70	1.70	1.27	1.00
C1_BOW_DD	99.30	2.18	19.46	13.29	9.70
C1_BOW	100.00	2.60	17.69	11.02	8.23
C2_SOW_DD	100.00	1.50	1.50	1.19	1.00
C2_SOW	100.00	1.59	1.59	1.23	1.00
C2_BOW_DD	98.39	2.26	30.22	20.66	15.06
C2_BOW	98.55	2.59	16.17	9.79	7.06
C3_SOW_DD	100.00	1.53	1.53	1.21	1.00
C3_SOW	100.00	1.93	1.93	1.36	1.00
C3_BOW_DD	97.27	2.79	35.43	22.57	17.25
C3_BOW	99.07	2.58	21.08	13.58	10.12

4.4 Analysis and Discussion

Summarizing the results in our experiments, the following analysis is given:

- In the setting of SOW, the influence of architectures on adversarial attacks is small. However, in the setting of BOW, the classifiers that have more layers and neurons are normally less vulnerable to adversarial examples. How the architectures influence the adversarial attacks is still unknown. This will be investigated in our future work.
- Different feature extraction methods have different abilities to resist adversarial examples. In our experiments, the BOW-based classifiers are less vulnerable to adversarial examples than the SOW-based classifiers. This implies that the robustness of deep learning based systems in the field of host intrusion detection can be improved by choosing robust feature extraction methods. How these methods influence the robustness of the target models against adversarial examples is not discussed in this paper. We leave this to future work.
- The adversarial training shows better defense performance than defensive distillation in our experiment. Especially, the combination of BOW and adversarial training reduces the success rate by 4.15 percent on average.
- Among all the l_p norms, we believe l_1 norm is the best one to measure the distances between original inputs and corresponding adversarial examples in the field of host intrusion detection. Because it intuitively indicates the number of system calls added to original system call traces.
- Adversarial attack methods just need to modify a very small but key part of features in original inputs to generate the adversarial examples which can mislead the DL-HIDS.
- ISM outperforms the state-of-the-art adversarial attack methods in the domain of host intrusion detection. Although defensive mechanisms such as adversarial training and defensive distillation can decrease the success rate of ISM against the target classifiers, most adversarial examples generated by ISM still evade the detection of the target classifiers reinforced by these defensive mechanisms. This further demonstrates the effectiveness of our method.

5 Conclusions

DL-HIDSs are vulnerable to adversarial examples. In this paper, to better evaluate the robustness of DL-HIDS against adversarial examples, we propose a new adversarial attack method known as the iterative step method to generate adversarial examples against DL-HIDS. The results show ISM is practical in the domain of host intrusion detection. It can be used by the creators of the target models to evaluate the robustness of their models before publishing.

Sometimes, the adversaries can not access the internal parameters of the target classifiers. In this setting, attacks can be only performed in a black-box way, which is not discussed in this paper. Future work will attempt to investigate the black-box attacks against deep learning based systems in the field of cybersecurity. In this paper, we only explore the attacks against DFNN-based HIDS. In the future, we will extend ISM to other neural networks such as recurrent neural networks. We are going to explore the attack performance of ISM against the other deep learning based systems in cybersecurity such as malware detection systems.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant U1831131, in part by the Central Government Guides Local Science and Technology Development Special Funds under grant [2018]4008, in part by the Technology Cooperation Key Project of Guizhou Province, China under grant [2015]7763, and in part by the Science and Technology Planned Project of Guizhou Province, China under grant [2020]2Y013. I would like to give my most sincere thanks to Prof. Xie, Xiaoyao and Prof. Xu, Yang for their generous assistance and valuable suggestions.

References

[1] M. Abadi, P. Barham, J. Chen, *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, pp. 265-283, 2016.

[2] A. S. Abed, T. C. Clancy, and D. S. Levy, "Applying bag of system calls for anomalous behavior detection of applications in linux containers," in *IEEE Globecom Workshops (GC Wkshps'15)*, pp. 1–5, 2015.

[3] P. Agrawal and S. Ganapathy, "Modulation filter learning using deep variational networks for robust speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 244–253, 2019.

[4] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[5] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of IEEE Symposium on Security and Privacy (SP'17)*, pp. 39–57, 2017.

[6] A. Chawla, B. Lee, S. Fallon, and P. Jacob, "Host based intrusion detection system with combined cnn/rnn model," in *Joint European Confer-*

ence on Machine Learning and Knowledge Discovery in Databases, pp. 149–158, 2018.

[7] B. Chen, Z. Ren, C. Yu, I. Hussain, and J. Liu, "Adversarial examples for cnn-based malware detectors," *IEEE Access*, vol. 7, pp. 54360–54371, 2019.

[8] J. Clements, Y. Yang, A. Sharma, H. Hu, and Y. Lao, "Rallying adversarial techniques against deep learning for network security," *Cryptography and Security*, 2019. arXiv:1903.11688. (https://arxiv.org/pdf/1903.11688.pdf).

[9] G. Creech and J. Hu, "Generation of a new ids test dataset: Time to retire the kdd collection," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'13)*, pp. 4487–4492, 2013.

[10] A. Dewanje and K. A. Kumar, "A new malware detection model using emerging machine learning algorithms," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 24–32, 2021.

[11] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'18)*, pp. 9185–9193, 2018.

[12] T. T. Gao, H. Li, and S. L. Yin, "Adaptive convolutional neural network-based information fusion for facial expression recognition," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 17-23, 2021.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR'15)*, 2015. arXiv:1412.6572.

[14] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial examples for malware detection," in *Proceedings of European Symposium on Research in Computer Security*, pp. 62–79, 2017.

[15] Y. He, "Identification and processing of network abnormal events based on network intrusion detection algorithm," *International Journal of Network Security*, vol. 21, no. 1, pp. 153–158, 2019.

[16] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *Computer Vision and Pattern Recognition*, 2016. arXiv:1607.02533.

[17] H. Li, S. Zhou, W. Yuan, J. Li, and H. Leung, "Adversarial-example attacks toward android malware detection system," *IEEE Systems Journal*, vol. 14, no. 1, pp. 653–656, 2020.

[18] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," *Cryptography and Security*, 2018. arXiv:1809.02077.

[19] X. Liu, X. Du, X. Zhang, Q. Zhu, H. Wang, and M. Guizani, "Adversarial samples on android malware detection systems for IoT systems," *Sensors*, vol. 19, no. 4, pp. 974, 2019.

[20] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, pp. 2574–2582, 2016.

[21] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proceedings of IEEE European Symposium on Security and Privacy*, pp. 372–387, 2016.

[22] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proceedings of IEEE Symposium on Security and Privacy (SP'16)*, pp. 582–597, 2016.

[23] D. Sierra-Sosa, B. Garcia-Zapirain, C. Castillo, I. Oleagordia, R. Nuno-Solinis, M. Urtaran-Laresgoiti, and A. Elmaghraby, "Scalable healthcare assessment for diabetic patients using deep learning on multiple gpus," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5682–5689, 2019.

[24] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[25] Y. Sun, S. Yin, and L. Teng, "Research on multi-robot intelligent fusion technology based on multi-mode deep learning," *International Journal of Electronics and Information Engineering*, vol. 12, no. 3, pp. 119–127, 2020.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, pp. 1–9, 2015.

[27] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *Computer Vision and Pattern Recognition*, 2013. arXiv:1312.6199.

[28] A. Tewari, M. Elgharib, G. Bharaj, F. Bernard, H. Seidel, P. Perez, M. Zollhofer, and C. Theobalt, "Stylerig: Rigging stylegan for 3D control over portrait images," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6141–6150, 2020.

[29] J. X. Tong, H. Li, and S. L. Yin, "Research on face recognition method based on deep neural network," *International Journal of Electronics and Information Engineering*, vol. 12, no. 4, pp. 182–188, 2020.

[30] Z. Wang, "Deep learning based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367–38384, 2018.

[31] C. Xu, J. Shen, and X. Du, "A method of few-shot network intrusion detection based on meta-learning framework," *IEEE Transactions on Infor-*

mation Forensics and Security, vol. 15, pp. 3540–3552, 2020.

[32] K. Yang, J. Liu, C. Zhang, and Y. Fang, "Adversarial examples against the deep learning based network intrusion detection systems," in *Proceedings of IEEE Military Communications Conference (MILCOM'18)*, pp. 559–564, 2018.

[33] S. Y. Yerima and S. Sezer, "Droidfusion: A novel multilevel classifier fusion approach for android malware detection," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 453–466, 2019.

[34] S. C. Zhang, X. Y. Xie, and Y. Xu, "Intrusion detection method based on a deep convolutional neural network," *Journal of Tsinghua University (Science and Technology)*, vol. 59, no. 1, pp. 44–52, 2019.

Biography

Sicong Zhang was born in Chongqing, China. He received the B.E. degree in electrical engineering and automation from the Civil Aviation University of China and the M.E degree in computer science and technology from Guizhou Normal University. He received the Ph.D. degree in software engineering at Guizhou University, Guiyang, China. His research interests include cybersecurity, deep learning, and optimization theory.

Xiaoyao Xie was born in Guizhou, China. He is now a professor and doctoral supervisor with Key Laboratory of Information and Computing Science Guizhou Province, Guizhou Normal University, Guiyang, China. He is also the director of Key Laboratory and the vice present of Guizhou Normal University. His research interests include IPv6, 5G, and cybersecurity.

Yang Xu was born in Shandong, China. He received his Ph.D. in computer software and theory from Guizhou University. He is now a professor and postgraduate supervisor with Key Laboratory of Information and Computing Science Guizhou Province, Guizhou Normal University, Guiyang, China. He is a Senior Member of the China Computer Federation (CCF). His research interests include cybersecurity and machine learning.

A Confidential Information Hiding Scheme for 3D Model Based on Contour Analysis

Shuai Ren, Aoxiong Fan, Lei Shi, Xuemei Lei, and Zhuoyi Dan
(Corresponding author: Aoxiong Fan)

School of Information Engineering, Chang'an University
Middle-section of Nan'er Huan Road,Xi'an 710064, China
Email: fanaoxiong@chd.edu.cn

(Received Mar. 26, 2020; Revised and Accepted Dec. 10, 2020; First Online June 1, 2021)

Abstract

3D model-based information hiding is an important topic in carrier-based confidential information sharing. In order to satisfy the requirements of confidential communication such as imperceptibility, robustness and capacity, an information hiding algorithm based on the interval analysis of values on z-axis for the vertical contour curves of 3D models. First, the 3D models will be scaled disproportionately and then rotated. Both the scaling and rotation are used to help the models fit a rectangle with fixed dimensions. And the vertical contour curves can be obtained by horizontal mapping. Second, the vertical contour curves will be mapped into the two-dimensional coordinate system and the interval distance values on the vertical axis can be preset as a fixed number. And then some vertical coordinate values will be picked out by the fixed interval distance. Last, one vertical coordinate value will be considered as the midpoint of a certain range, and all of the vertical coordinate values in this range will be changed into the same as the midpoint, and then will be converted into the binary numbers. From the theory analysis and experiment, it can be seen that the algorithm can effectively resist the scaling attack. According to the fixed rotation angle and the interval distance, the data can be embedded repeatedly as the in redundancy, which means the confidential information can be scattered into the whole model and the algorithm can be robust against cutting. Specifically, this algorithm is of strong robustness against the random noise under 0.2%, remeshing and non-uniform simplification.

Keywords: 3D Mesh; Contour Analysis; Information Hiding; Two-Dimensional Mapping

1 Introduction

As an important branch of information security, information hiding technology hides secret information in multimedia carriers, so that the secret carrier has the ability to resist external attacks while meeting the characteristics of HVS, and realize the purpose of covert communication. Currently, information hiding algorithms based on digital images have achieved more research results [2]. With the rapid development of computer graphics technology and computer software and hardware technology, more and more three-dimensional models are available on the Internet. With its rich visual details, the three-dimensional model is more used in information hiding research, which promotes the innovation and breakthrough of information hiding algorithm research. At present, most of the important research topics on the security of information content are the research of information hiding algorithms based on digital images, while the research on information hiding technology based on 3D model carriers is less [3]. At present, with the development of 3D movies to 3D printing technology and the support of hardware devices, there are more and more 3D models on the Internet, which will surely be the main form of future data. It is necessary to study information hiding technology based on 3D models.Starting from the pre-processing of 3D model in spatial domain, we are going to hide the confidential information by modification on the geometric data of the model. It can be seen that hiding information based on the spatial modification is simple and direct, but it is always a problem to ensure the performance. Many researchers have contributed for the performance improving of 3D model-based information hiding in spatial domain.

Tsai *et al.* [12] used a spatial encoding method with embedding threshold to embed secret information into the encrypted vertices of the three-dimensional model to realize information hiding; Zhang *et al.* [15] calculated the distance between two vertices on the three-dimensional model. The coordinate difference constructs the difference histogram, and at the same time, in order to increase the embedding capacity, a multi-level histogram shift mechanism is introduced; Li *et al.* [8] use the local geometric features of the 3D model to embed secret information, including local object curvature, vertex method the local geometric shape features in the linear and spherical coordinate systems make the algorithm invisible; Tsai *et al.*

used PCA (Principal Component Analysis) in the literature [13] to calculate the surface of the three-dimensional model Complexity, adaptive secret information embedding, while using a constant threshold to control the maximum embedding capacity of each vertex of the model to reduce distortion; Wang proposed in [14] a method that can deal with affine transformation. The dense image is embedded in the distance from the vertex to the center of the model, which effectively improves the robustness of the algorithm.

All of the above research works are executed according to the traditional operand rules in order to improve some performance. Considering the poor feasibility of 3D carrier processing in transform domain, most of the preprocessing methods for 3D carrier are spatial domain-based. Information hiding technologies based on the optical theories are multi-dimensional [6,9], with large capacity and strong robustness, and are mostly operated in spatial domain. Research on 3D model analysis in spatial domain is of practical importance, such as the 3D models generation in computer integrated imaging system based on the intelligent depth inversion model. Additionally, the structure feature analysis and carrier preprocessing for 3D models in our research are theoretically based on the spatial domain operations.

By analyzing the advantages and disadvantages of spatial domain operations,a new 3D mesh carrier-oriented information hiding algorithm is proposed. Taking the contour of 2-dimensional space by horizontal rotation of the model, the change of tangent line angle between positive and negative for some vertices on the contour will be represented as 0/1 data. Here, these vertices are selected according to a certain interval value on the horizontal axis. Adopting a certain rotation angle and a certain horizontal axis interval, the redundancy embedding will be achieved. Obviously, the geometrical attacks cannot change the change of tangent line angle between positive and negative, so the robustness against the geometrical attack will be good. The performance of this algorithm will synthetically depend on the horizontal rotation angle of 3D model, the horizontal interval value and the details of redundancy strategy.

2 Information Hiding Based on the 3D Model Vertical Contour Analysis

The information hiding algorithm (SO, Shadow Outline) proposed in this paper based on the statistics of the inflection point of the longitudinal contour of the 3D model is divided into 11 steps, including the preprocessing of the 3D model carrier and the process of embedding secret information. The information hiding algorithm flowing in this paper is shown in Figure 1.

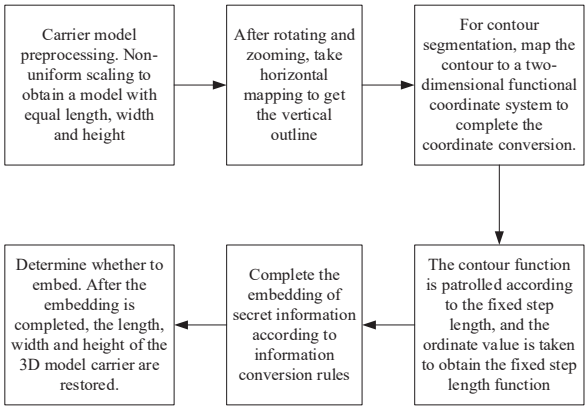


Figure 1: Information hiding algorithm flow in this article

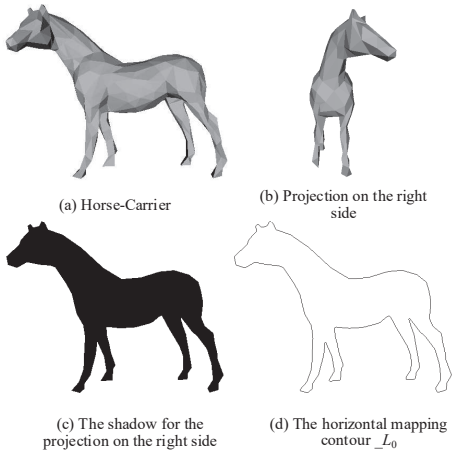


Figure 2: The horizontal mapping process

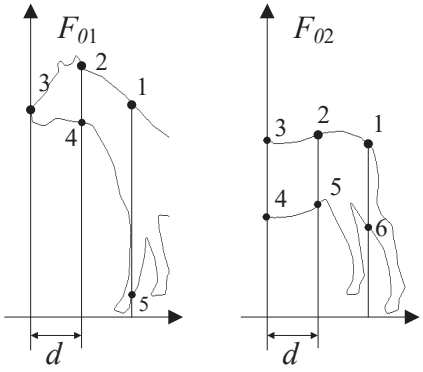
Step 1: The file for the 3D model will be read as the carrier, and the model will further be non-uniform scaled to fit a rectangle with fixed dimensions. The value for the length, width and height of the model is given as G.

Step 2: The model in Figure 1 will be horizontally placed, and mapped in 2-dimensional to form a contour represented as L_α. Here,α is the rotation angle horizontally, and the contour generation process when $\alpha = 0^o$ is shown as Figure 2(a)-(d).

Step 3: The contour will be divided into two parts from its horizontal midpoint. Being projected when the rotation angle is α, the contours for the two parts are represented as L_{α_1} and L_{α_2}.For example, L_0 is the contour when $\alpha = 0^o$, and the two parts divided from L_0 can be represented as L_{0_1} and L_{0_2}, which are shown in Figure 3.

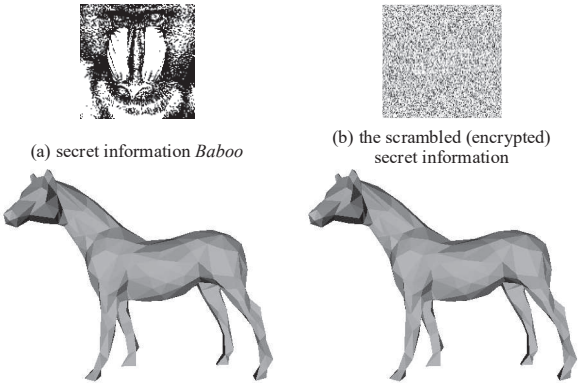
Step 4: The coordinates for L_{α_1} and L_{α_2} will be transformed into 2-dimensional form and denoted as two functions F_{α_1} and F_{α_2}. For example, L_{0_1} and L_{0_2} in Step3 will be transformed as F_{0_1} and F_{0_2}.

Step 5: Some vertical coordinate values of F_{α_1} and F_{α_2} will be selected according to the given interval value



(a) Contour L_{01} (b) Contour L_{02}

Figure 3: Horizontal mapping of 3D model



(a) secret information *Baboo* (b) the scrambled (encrypted) secret information

(c) the original model Horse (d) the stego model Horse

Figure 4: The information process schematic

d on x axis, these two sets of selected vertical coordinated values can be denoted as D_{α_1} and D_{α_2}. For example, the selected vertical coordinated values of L_{0_1} and L_{0_2} are respectively D_{0_1} and D_{0_2}.

Step 6: D_{α_1} and D_{α_2} will be transformed into another data form according to a given rule named interval transform which is shown in Table 1. K is used to represent the interval repeat times, $k \in \mathbf{Z}^+$, and e is the density threshold for data division, $2 \leq e \leq G/2$.

Table 1: Data transformation rule

Interval threshold	The data after transformed
$[kG/e, (k+1)G/e)$	-1
$[(k+1)G/e, (k+2)G/e]$	1

Step 7: According to the information expression rules in Table 1, corresponding to the D_{α_1} and D_{α_2} data, the internal information contained in the carrier before the modification can be parsed, and the information to be hidden is compared with its own information. If it is not the same, it will directly correspond to the D_{α_1} and D_{α_2} functions. The value is modified, that is, the information is converted according to the modification in Table 1, and then the converted value is normalized.So the same information has been embedded into D_{α_1} and D_{α_2} to achieve the redundancy embedding.

Step 8: The contour data will be analyzed from the left to the right on x axis for a distance of which the length is f (integral multiplies of d), and will be embedded repeatedly. The embedded data are respectively represented as $D_{\alpha_1\_h}$ and $D_{\alpha_2\_h}$, and the repeat time is represented by h.

Step 9: The model deformed from Step 1 will be rotated by β on Z axis.

Step 10: Determine whether the cumulative rotation on Z axis is no less than γ. If it is, the old information will be repeatedly embedded, or else execute from Step1 to Step10.

Step 11: The length, width and height of the model embedded with secret data will be recovered to the way they were.

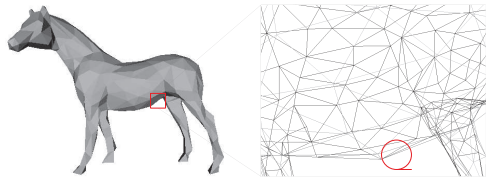
3 Test and Analysis

This article chooses Figure 4(a) for the secret information (128×128 binary image), Figure 4(b) for the secret information after preprocessing methods such as scrambling. Choose the three-dimensional model Horse shown in Figure 4(c) as the carrier. The three-dimensional model carrier is composed of 362 vertices and 720 faces,while the model shown in Figure 4(d) is the model with secret information when G=512, d=4, f=32, e=128, $\beta = 5^o$ and $\gamma = 90^o$. The experiment was operated based on VC++, OpenGL and Matlab.

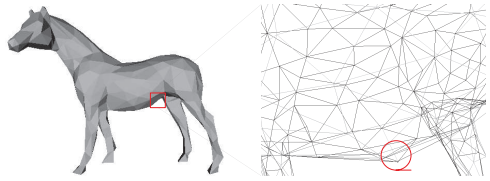
3.1 Imperceptibility and Capacity

1) Based on HVS: According to the section modification strategy in Table 1, the maximum modified value is the distance between two pixels, when G=512 and e=128. Whether the visual effect of the distance between two pixels or the transmission effect of those two pixels is vulnerable to be ignored, as a result modification based on the distance between two pixels is good for the imperceptibility. The local enlarged view of the model with modified pixel distance is shown in Figure 5, which can satisfy HVS property.

2) Based on Hausdorff distance: Using Hausdorff distance [1], the maximum mismatch of two point sets is used to measure the invisibility of the algorithm, thereby quantifying the invisibility index.The imperceptibility can be quantized by Hausdorff distance. A toolkit named Metro will be used to calculate



(a) The original model-Horse



(b) The stego model-Horse

Figure 5: The information process schematic

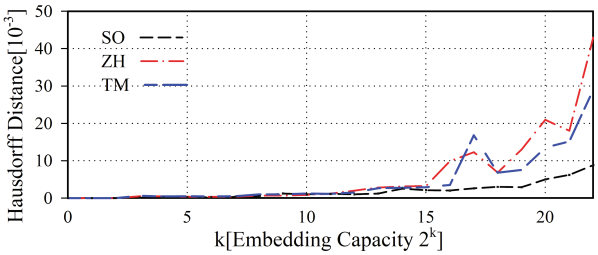


Figure 6: Imperceptibility/Capacity(Hausdorff Distance-k)

the Hausdorff distance for stego model Horse embedded with Baboon.The Hausdorff distance is 0.000781 when embedding data is 216 bit.

The comparative experiment of imperceptibility for SO-based algorithm by Hausdorff distance is shown in Figure 6, in which the horizontal ordinate is the embedding quantity 2^k,and the vertical ordinate is the Hausdorff distance.ZH and TM represent the Zero High Resolution information hiding algorithm [10] and the algorithm based on 3D Model Texture Map [11] proposed by the research group.From Figure 6, when $k < 15$, the Hausdorff distance for SO-based algorithm is much less than it of ZH-based and TM-based algorithm. While when $k \geq 15$, the Hausdorff distance for SO-based algorithm is still much less than it of ZH-based algorithm and a little bit less than it of TM- based algorithm. It can prove that the imperceptibility is better than it of ZH-based and TM-based algorithm, as well as the capacity.

The skeleton similarity matching: The comparative experiment based on skeleton similarity E^n for the imperceptibility is shown in Figure 7, in which the similarity of ZH-based algorithm is better than it of TM-based algorithm. And when $k \geq 8$, the similarity of SO-based algorithm is much better than it of ZH-based algorithm. It can prove that SO-based algorithm is better than other two algorithms on imperceptibility especially when the embedding capacity was increased.

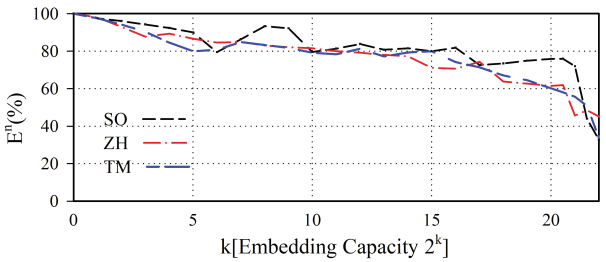


Figure 7: Imperceptibility/Capacity($E^n - k$)

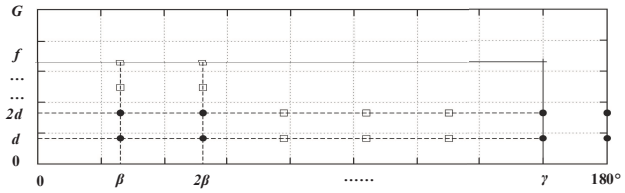


Figure 8: Redundancy embedding schematic of SO-based algorithm

3.2 Robustness

The secret information will be hided into the ordinate value of the model projection contour. Actually the secret information will be embedded into other two coordinate values redundantly shown as Figure 8.

The secret information will be actually embedded into two regions such as γ and $f.G$ is the multiple of f, and γ is the multiple of β, while 180 is the multiple of γ, as a result the secret information will be averagely distributed into the whole model redundantly. The stego model consequently can be robust against the cropping attack of which the degree is lower than $[100(512\times360) - (\gamma\times f)/(512\times360)\%]$.The robustness experiments against other attacks are shown in Figures 8, 9 and 10, and some related mathematical parameters about robustness are

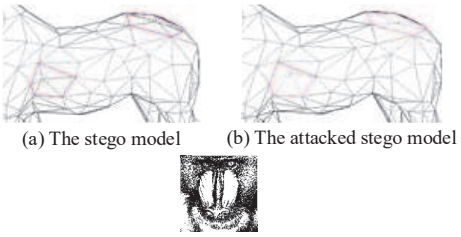
1) The BER (bit error rate) of the extracted information bit sequence represents the error bit rate of the extracted information, and the calculation is shown in Equation (1), Where Ψ represents the number of error bits in the extracted information, and Θ is the total number of bits embedded in the secret information;

2) The correlation coefficient [7] between the extracted sequence $\{x_i\}$ and the original sequence $\{y_i\}$, which can be represented by Equation (2).

Here, $\overline{x}$ and $\overline{y}$ are the average values of $\{x_i\}$ and $\{y_i\}$ respectively.

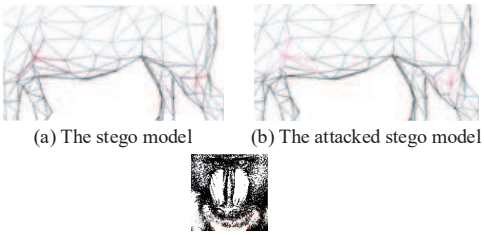
$$\text{BER} = \frac{\Psi}{\Theta} \tag{1}$$

$$\text{Corr} = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\left[\sum_{i=1}^{n}(x_i-\overline{x})^2\sum_{i=1}^{n}(y_i-\overline{y})^2\right]^{1/2}} \tag{2}$$


(a) The stego model (b) The attacked stego model

(c) The extracted information

Figure 9: Random noise (0.1%)


(a) The stego model (b) The attacked stego model

(c) The extracted information

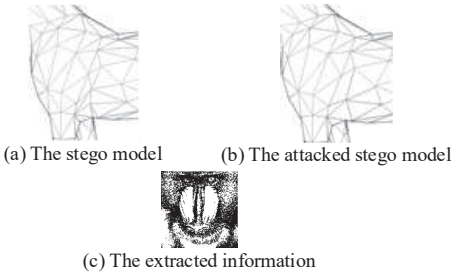
Figure 10: Remeshing

It is really robust against random noise (the intensity of which is less than 0.1%), re-meshing and uniform simplification, shown in Figures 9, 10 and Figure 11. The extraction information is shown as Figure 9(c), Figure 10(c) and Figure 11(c), respectively. The relationship between the intensity of random noise and BER, and the relationship between the intensity of random noise and Corr are shown in Figures 12 and 13, which can indicate the good robustness.

Similarly,the relationship between the intensity of uniform re-meshing and BER, and the relationship between the intensity of uniform re-meshing and Corr are shown in Figures 14 and 15 shown, which can indicate the good robustness.

The relationship between the intensity of uniform simplification and BER, and the relationship between the intensity of uniform simplification and Corr are shown in Figures 16 and 17 shown, which can indicate the good robustness.

The robustness of the algorithm proposed in this paper is compared with the literature [4] and [5]. In the literature [4], the three-dimensional model is converted


(a) The stego model (b) The attacked stego model

(c) The extracted information

Figure 11: Uniform simplification

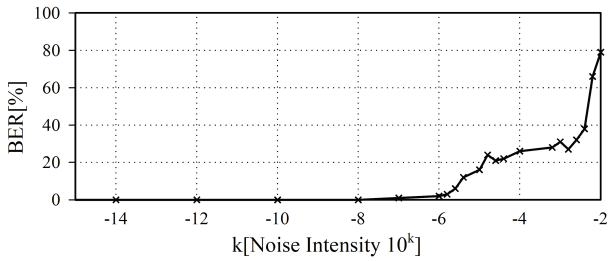


Figure 12: The Robustness against Random noise(BER)

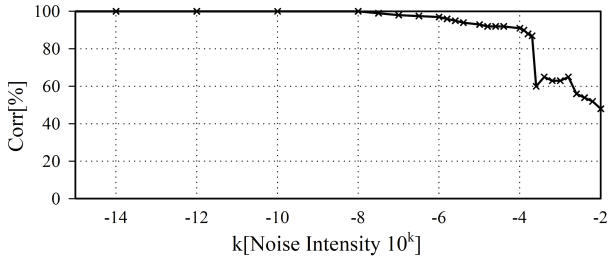


Figure 13: The Robustness against Random noise (Corr)

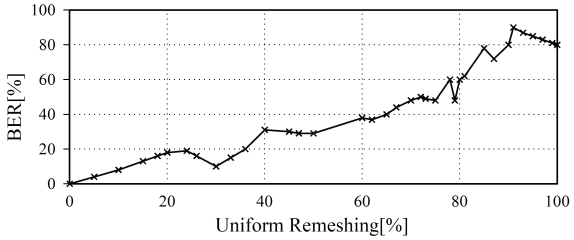


Figure 14: The robustness against Uniform Re-meshing (BER)

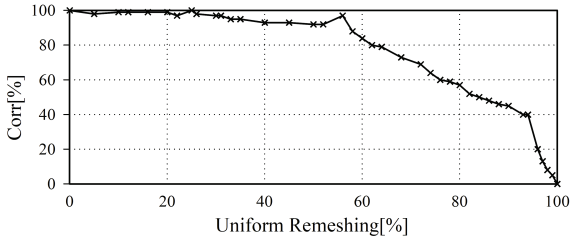


Figure 15: The robustness against Uniform Re-meshing (Corr)

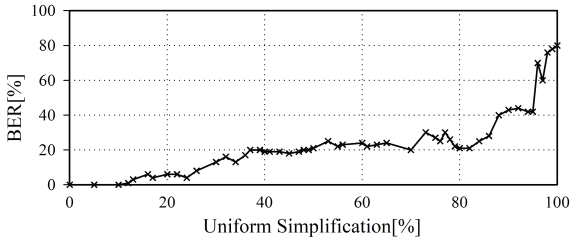


Figure 16: The robustness against Uniform simplification (BER)

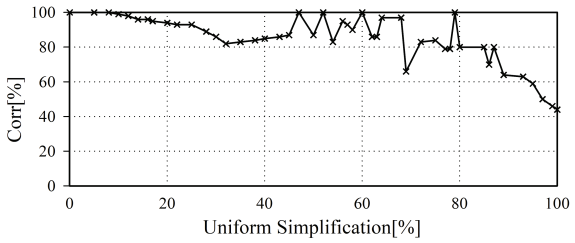


Figure 17: The robustness against Uniform simplification (Corr)

into a two-dimensional image, the two-dimensional image is transformed, and the secret information is embedded in the image. In [5], the distance normalized modulation method is used, and the secret information is embedded in the model by modulating the average value of the normalized distance. It can be seen through experimental comparison that the robustness of the algorithm is greatly improved compared with it, as shown in Figure 18.

From Figure 19, the robustness of SO-based algorithm is much better than it of other algorism in paper [4] and [5] against uniform re-meshing.

It is shown in Figure 20 that the robustness of SO-based algorithm is much more improved than it of these contrast algorithms against uniform simplification.

3.3 Complexity

Figure 21 is a comparison experiment of computational complexity based on SO and literature [4] algorithm and

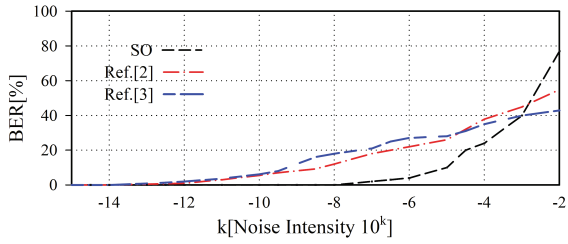


Figure 18: The comparison of robustness for SO-based and other algorithms (Noise-BER)

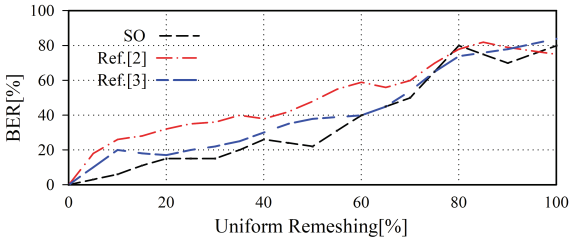


Figure 19: The comparison of robustness for SO-based and other algorithms (Uniform Remeshing-BER)

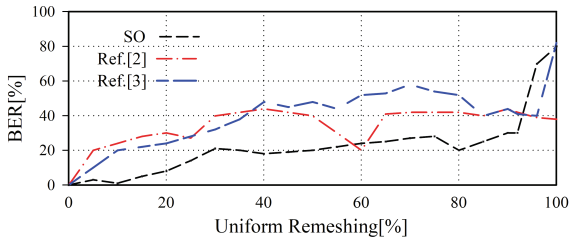


Figure 20: The comparison of robustness for SO-based and other algorithms (Uniform Simplification-BER)

literature [5] algorithm. It can be seen that when the embedding amount is the same, the calculation time ratio of SO-based algorithm is compared with the algorithm of literature [4] and literature [5]. The computing time is short.

4 Conclusions

From the above, the vertical mapping was used to analyze the model contour. After that, the values of z were divided into different groups according to their value. Then a 0/1 sequence was obtained and will be treated as the embedded object to cover the secret data. Additionally, the non-proportional scaling and the redundantly embedding strategy were really effective to improve the robustness against cropping. From the comparison experiment, this algorithm is better in robustness than ZH-based and TM-based algorithms, and much better in complexity compared with other similar algorithms. Conclusively, this

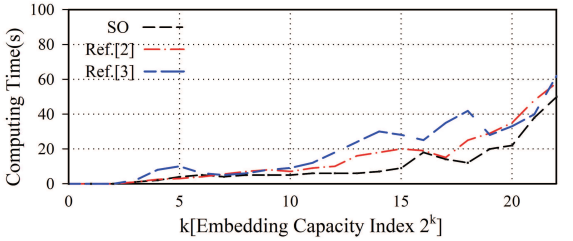


Figure 21: The comparison of time complexity experiments between SO and other algorithms (Computing Time-k)

algorithm was suggested to be applied to satisfy the high demand of robustness.

Acknowledgments

The research has been supported by National Natural Science Foundation of China (61702050); "Double First Class" Guiding Project of the Central University of China (300104292405); The Fundamental Research Funds for the Central Universities, CHD (300102240208); The 2019 Chang'an University Graduate Education and Teaching Reform Construction Special Infrastructure Construction Project (300103190640); College Students' Innovation and Entrepreneurship Training Program (201910710079, 201810710052, 201810710060, 201810710215, 201810710224).

References

[1] N. A. Carlson and J. R. Porter, "On the cardinality of hausdorff spaces and h-closed spaces," *Topology and its Applications*, vol. 160, no. 1, pp. 137–142, 2017.

[2] C. C. Chang, K. F. Hwang, M. S. Hwang, "Robust authentication scheme for protecting copyrights of images and graphics", *IEE Proceedings-Vision, Image and Signal Processing*, vol. 149, no. 1, pp. 43-50, 2002.

[3] T. Y. Chen, M. S. Hwang, and J. K. Jan, "Adaptive authentication schemes for 3D mesh models," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 4561-4572, 2009.

[4] A. Dong and R. Zeng, "Research and implementation based on three-dimensional model watermarking algorithm," in *International Conference on Computing Intelligence and Information System*, 2017. DOI: 10.1109/CIIS.2017.47.

[5] X. Feng, "A watermarking for 3D point cloud model using distance normalization modulation," in *The 4th International Conference on Computer Science and Network Technology (ICCSNT'16)*, 2016. DOI: 10.1109/ICCSNT.2015.7491001.

[6] C. Ji, H. Deng, and Q. Wang, "Pixel extraction based integral imaging with controllable viewing direction," *Journal of Optics*, vol. 14, no. 9, pp. 095401, 2012.

[7] Z. Li, A. G. Bors, "Steganalysis of 3D objects using statistics of local feature sets," *Information Sciences*, vol. 415-416, pp. 85-99, 2017.

[8] J. Liu, Y. Yang, D. Ma, Y. Wang, and Z. Pan, "A watermarking method for 3D models based on feature vertex localization," *IEEE Access*, vol. 6, pp. 566122–56134, 2018.

[9] Y. Liu, X. Yang, and P. Luo, "New technology of 3D spatial digital watermarking based on integrated imaging," in *Proceedings of the 11th National Conference on Information Hiding and Multimedia Information Security*, pp. 167–172, 2014.

[10] S. Ren, M. Wang, A. Fan and Z. Gao, J. Xu, S. Khurram, and T. Z. Tao, "An information hiding algorithm for zero high resolution 3D mesh model," *Computer Science*, vol. 47, pp. 328–334, 2020.

[11] S. Ren, Z. Wang, Z. Xu, D. Su, and Y. He, "An information hiding algorithm based on texture mapping of OBJ three-dimensional model," *Journal of Beijing University of Posts and Telecommunications*, vol. 42, pp. 22–27, 2019.

[12] Y. Y. Tsai, "Separable reversible data hiding for encrypted three-dimensional models based on spatial subdivision and space encoding," *IEEE Transactions on Multimedia*, no. 99, pp. 1–1, 2020.

[13] Y. Y. Tsai, W. C. Huang, and B. F. Peng, "An efficient and distortion-controllable information hiding algorithm for 3D polygonal models with adaptation," *International Journal of Network Security*, vol. 17, no. 1, pp. 79–84, 2015.

[14] Q. Wang, X. Feng, and S. Yan, "A digital watermarking algorithm against affine transformation for 3D mesh model," in *The 5th International Conference on Computer Science and Network Technology (ICCSNT'16)*, 2016. DOI: 10.1109/ICCSNT.2016.8070264.

[15] Q. Zhang, T. Wen, and X. Song, "Multilevel reversible data hiding based on difference histogram for 3D point cloud models," in *The 6th International Conference on Information Science and Control Engineering (ICISCE'19)*, 2019.

Biography

Shuai Ren received his Ph.D. degrees in Computer Science and Technology from Northwestern Polytechnical University,Xi'an,Shaanxi,in 2010 and his B.S.degree in Information Confrontation from Northwestern Polytechnical University,Xi'an,Shaanxi, in 2005.He is an Associate Professor of School of Information Engineering,Chang'an niversity,Xi'an,Shaanxi,China,since 2010.His research interests include Information hiding and digital watermarking technology, digital image processing and 3D model processing.

Aoxiong Fan received the B.S. degree in Network Engineering from Xi'an University of Posts & Telecommunications, Xi'an, Shaanxi,in 2018.He is currently pursuing his M.S. degree in the School of Information Engineering, Chang'an University, Xi'an, China. His major is Computer Technology. His research interests include Information hiding, digital watermarking and 3D model simplification.

Lei Shi received the B.S. degree in software engineering from North University of China,Taiyuan,Shanxi,in 2019. He is currently pursuing his M.S. degree in the School of Information Engineering, Chang'an University, Xi'an, China.. His research interests include information hiding

and 3D model.

Xuemei Lei received the B.S. degree in Zhengzhou Normal University, Zhengzhou, Henan,in 2019.She is currently pursuing her M.S. degree in the School of Information Engineering, Chang'an University, Xi'an, China. Her research interests include information hiding based on multi carrier 3D model.

Zhuoyi Dan received the B.S. degree in communication engineering from Taiyuan University of Science and Technology, Taiyuan, Shanxi,in 2018. She is currently pursuing her M.S. degree in the School of Information Engineering, Chang'an University, Xi'an, China. Her research interests include information hiding technology.

A Survey on Membership Inference Attacks Against Machine Learning

Yang Bai[1,2], Ting Chen[1], and Mingyu Fan[1]
(Corresponding author: Mingyu Fan)
School of Computer Science and Engineering, University of Electronic Science and Technology of China, China[1]
No.2006, Xiyuan Road, West High Technology District, Chengdu 611731, China
No.30 Institute of CETC, China[2]
No.8, Chuangye Road, High Technology District, Chengdu 611731, China
Email:alicepub@163.com, brokendragon@uestc.edu.cn, ff98@163.com
(Received May 31, 2020; Revised and Accepted Feb. 10, 2021; First Online June 2, 2021)

Abstract

Nowadays, machine learning is widely used in various applications. However, machine learning models are vulnerable to various membership inference attacks (MIAs) that leak information on the individual records trained by these models. Although many studies focus on finding new attack methods or improving attack performance, how to characterize MIAs is not well studied. This paper focuses on MIAs and the defense mechanisms against them by analyzing a framework that allows the general decomposition of existing MIAs against machine learning systems. We investigate MIAs by multiple key elements related to the victim model, including the adversary's observation, the prior knowledge of attacks, the classification of the target model, and the learning frame of the target model. Then, we classify the adversary's prior knowledge into seven sub-classes to further analyze the existing attacks. After that, we survey defense mechanisms employed by existing models. Our work contributes to understanding: 1) What is the working mechanism of MIAs; 2) Which components should be considered during the design of an MIA.

Keywords: Analysis Framework; Defense; Membership Inference Attack; Machine Learning

1 Introduction

In recent years, machine learning has been widely used in privacy-sensitive applications, *e.g.*, image recognition [16,25,33,59], speech recognition [21], healthcare data management [6, 14].

In such applications, privacy threats should be considered when devising machine learning techniques, especially that the training data should be protected from leakage because the training data contains sensitive information such as patients' healthcare information, personal preference, personal photos. Recently, academic work has revealed a variety of privacy threats against machine learning.

Privacy issues [8,29–31,64,65] of machine learning techniques include model inversion [15, 24, 49, 54], model extraction [47,60], and membership inference [7,22,32,38,45, 46, 52, 53, 57, 62]. A model inversion attack tries to reconstruct the model's input from output information [15]. *e.g.* Fredrikson *et al.* [15] introduce a model inversion attack that infers sensitive features used as inputs to decision tree models. In a model extraction attack, an adversary obtains black-box access to one target model and attempts to learn a model that closely approximates to, or even matches the target model [60]. The malicious user can leverage a model extraction attack to avoid query charge from the machine learning service company. The membership inference attacker aims to infer whether a specific data record is in the training data set of the target model or not [57]. Such attack can leak the privacy of the training dataset. In this paper, we focus on the overvoew of membership inference attack (MIA) against machine learning.

MIA has been extensively studied in other research fields, such as genomics privacy [28, 54] and mobility privacy [49]. Shokri *et al.*'s [57] was the first work to apply MIA against machine learning. Since then, many MIAs were proposed [7, 22, 53]. ML-Leaks [53] proposed a generic attack by relaxing some assumptions to show that such attacks are very broadly applicable at LOGAN [22] and GAN-Leaks [7] propose MIAs towards the generative machine learning model.

Although many works aimed at analyzing privacy threats and defense in machine learning systems, there lacks studies about systematical analysis of MIA and comprehensive comparisons among various attacking approaches. Such an empirical study can help researchers to understand how these attacks happen, what constraint conditions these attacks face, and what capabilities the

attackers possess. With these motivations, we provide a general survey of MIA against machine learning. Our work contributes to understanding why MIA chooses the existing designs, what are the causal factors of MIA, and how is the researching progress of defensive methods.

In this work,firstly, we construct a comprehensive framework to analyze existing MIAs against machine learning systems, which concludes four aspects: The attack observations, the prior knowledge, the target model type, and the target frame. Then we conduct a deep analysis of the prior knowledge of existing MIA and classify them into three categories and seven sub-classes. In addition, we summarize the factors that cause MIA, and classify existing defense mechanisms preventing MIA against machine learning into three categories. We also discuss their applicability to different MIA approaches and their effectiveness at mitigating these attacks. Finally, we envision three notable trends in the research on MIA methods and mitigation, which are worthy of in-depth studies in future.

The remainder of this paper is organized as follows. We discuss the attack model and state-of-the-art attack of membership inference attack against machine learning in Section 2. Then we propose the analytical aspects of this attack in Section 3.In Section 4, we summarize the factors influencing the attack and retrospect the exist defending mechanisms. This paper concludes in Section 6.

2 Terms and Prior Works Related to MIA

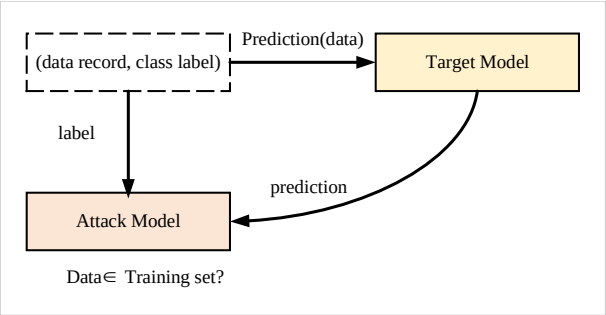


Figure 1: The working principle of MIA against machine learning

In this section, firstly, we introduce some terms that related to membership inference attack; Secondly, we give a brief review of prior works related to MIA by year-wise road map. The purpose of this section is to make us have a basic knowledge about MIA.

2.1 Terms

Membership inference attack (MIA). It shows in Figure 1 that Membership inference attacks aim to determine whether a given data point was present in the training data used to build a model [66]. Membership inference violates the privacy of both the individual participants involved in the model training and the owner of the training dataset [62]. This type of attack has been extensively studied in the adjacent area of genomics, and recently this attack is introduced in the context of machine learning [57]. In an MIA, the adversary attempts to infer whether a candidate data record is included in the training dataset of a target model. The adversary maybe given a candidate data record, or they can input some data point to target model and get out the query result. What's more, they might know some other background knowledge about the target model and training dataset. This attack then becomes a binary classification problem [57]. For each candidate record, there are two possible classes: The class "member" means that the candidate data is a member of the target model's training dataset, and the class "non-member" means otherwise. Thus, the adversary tries to establish a binary classifier to solve this problem.

Target model. In the MIA, the trained machine learning model will be treated as the adversary's target model.

Candidate data record. In the MIA, the candidate data records denote that a set of data sample which may belong to the target model's training dataset.

Shadow model. The shadow model is used to imitate the behavior of the target model, which is used in the black-box attack to obtain more information about the target model. During the attack, the adversary generates a shadow model by crafted shadow model training samples. Shadow models are models with the same architecture as the target model [45].

Attack model. Attack model is a binary classifier model used to infer the candidate data records whether are the member of the target model's training data. In other words, the adversary's attack process is the process of building an accurate attack model.

Machine Learning as a Service (MLaaS). MLaaS is an array of services that provide machine learning tools as part of cloud computing services. MLaaS helps clients benefit from machine learning without the cognate cost, time, and risk of establishing an in house internal machine learning team. Infrastructural concerns such as data pre-processing, model training, model evaluation, and ultimately, predictions, can be mitigated through MLaaS.

2.2 Prior Works Related to MIA

Figure 2 shows that the year-wise road map of MIAs against machine learning system. We describe the prior works ralated MIA by the timeline of this research direction.

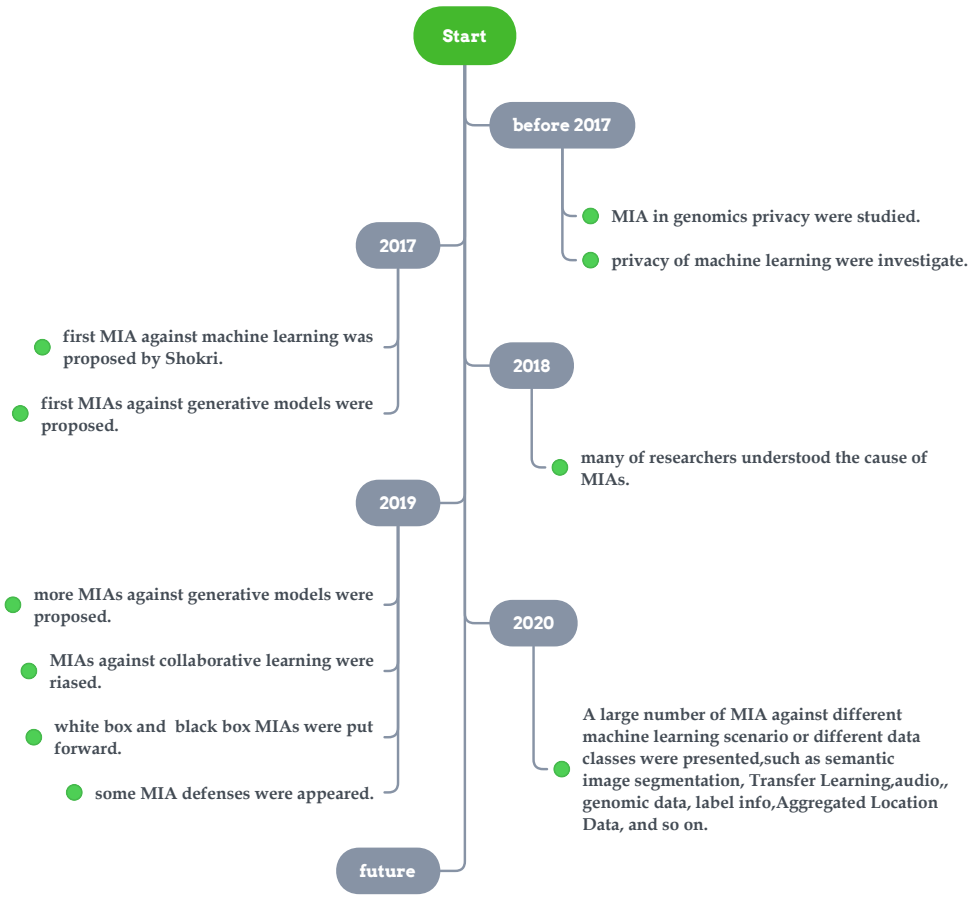


Figure 2: The year-wise road map of MIA against machine learning

Before 2017. Membership inference originated from genomics privacy related research [3, 28, 54], and then with the development of machine learning privacy research some related notions of privacy had appeared. Fredrikson *et al.* [15] demonstrated how the confidence information returned by many machine learning ML classifiers enables new model inversion attacks that could lead to unexpected privacy issues. Tramer *et al.* [60] explore model extraction attacks that could subvert model monetization, violate training-data privacy, and facilitate model evasion.

Year of 2017. 2017 should be the first year of MIA against machine learning.Because Shokri *et al.* [57] proposed the first MIA against machine learning, and they invented the shadow model technique to construct the attack models. as machine learning model includes discriminator and generator, Shokri *et al.*s' work only focused on MIA against discrimination model,but had not studied MIA in generation model. In LOGAN [22] the first MIA against generative models was presented. In this paper, Hayes *et al.* putted forward MIA against several state-of-the-art generative models, *e.g.*, Deep Convolutional GAN (DCGAN), Boundary Equilibrium GAN (BEGAN), and the combination of DCGAN with a Variational Autoencoder (DCGAN+VAE). The LOGAN introduces a full black-box attack model and a discriminator-accessible attack model against GANs. But the assumption of discriminator-accessible is the most knowledgeable but unrealistic setting because the discriminator in GAN is not always accessible in practice.

Year of 2018. Many researchers focused on understanding the cause of MIAs. Long *et al.*'s work [37]investigate and analyze membership attacks to understand why and how they succeed. And based on those understanding, they proposed Differential Training Privacy to estimate the privacy risk. In paper [38] reported a study that discover overfitting to be a sufficient but not a necessary condition for MIA to succeed, more specifically, they demonstrated that even a well-generalized model contains vulnerable instances subject. Yeom *et al.*'s [66] examined the effect that outfitting and influence have on the ability of an attacker to learn information about the training data from machine learning models, either through training set membership inference or attribute inference attacks. the cause factors will be discuss in Section 4.1. ML-Leaks [53] investigate the assumptions what a MIA requires. And they relaxes some assumptions of Shokri *et al.*'s work [57], such as the number of shadow models, the knowledge of the target model structure, and the target model's dataset information. This work reveals that attacks with relaxed assumptions are very broadly applicable at low cost and thereby pose a more severe risk than previ-

ously thought.

Year of 2019. In 2019, there were four main research advances in this direction. First of all, more MIAs against generative models were proposed. Hilpercht *et al.* [26] proposed a MIA based on Monte Carlo integration that applicable to all generative models. Chen *et al.* [7] explored the MIA against GANs and present the first taxonomy of MIAs in four classes, which included full black-box generator, partial black-box generator, withe-box generator and accessible discriminator. Secondly, MIAs against collaborative learning were raised. Melis *et al.* [41] proposed inference attacks against collaborative machine learning system. Nasr *et al.* [45] designed inference algorithms for both centralized and federated learning. thirdly, more white-box and black-box MIAs were put forward. Sablayrolles *et al.* [51] proposed a optimal inference strategy, the result showed that black-box attacks will perform as well as withe-box attacks in this optimal asympotic setting. Next, some MIAs defenses were appeared. Jia *et al.* [32] proposed MenGuard which adds noise to each confidence score vector by the target classifier to guarantees against black-box MIA. Rahimian *et al.* [50] studied the effect of Differential Privacy-Stochastic Gradient Descent(DP-SGD) to defense the MIAs. The systematic investigation of defenses against MIAs will be introduced in Section 4.

Year of 2020. A large number of MIA against different machine learning scenario or different data classes were presented. He *et al.* [23] show structural outputs of segmentation have severe risks of leaking membership, and present the first work on MIAs against semantic segmentation models while the prior works focus on classification models. As machine learning algorithms are used to process wireless signals, Shi *et al.* [55] presents how to leak privacy information from a wireless signal classifier by launching an over-the air MIA. Li *et al.* [36] investigate a MIA when the target model only provides the predicted label. Zhang *et al.* [68] propose LocMIA which allows adversaries to launch MIAs against aggregated LOCation data by train a binary classifier to infer whether a specific victim's location data involved in the aggregation group.

All of above, existing methods mainly study on finding out new attack approaches, improving the attack's performance, or proposing efficient mitigation methods against MIAs. But, none of the existing studies focus on the comprehensive analysis of MIA. Along with the development of machine learning privacy issues, more and more researchers pay attention to this field and the requirement of further research in this area increased. It is necessary to analyze the MIA from various angles to understand MIA better.

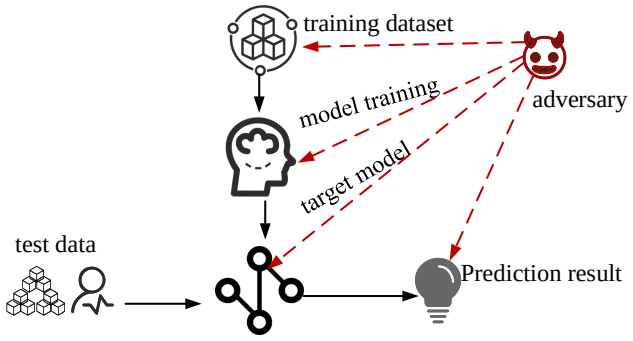


Figure 3: The elements considered by attacker to launch a MIA against machine learning

3 Analytical Framework of Membership Inference

3.1 Factors Considered in our Framework

As the processes of machine learning related to the general keywords include training dataset, model training, machine learning model, and prediction result. From the MIAs attackers view, their adversarial capability refers to the control-ability of these elements. In general, the adversary can condider several aspects for designing a MIA against machine learning system.

1) From the adversary's view, whether the adversary has knowledge about the training dataset, and what the training dataset background knowledge the attacker knows is the consideration elements.

2) During the model training, whether the model is a stand-alone or collaborative learning style, and whether the attacker is a bystander or one of the participants should be taken into consideration.

3) The adversary requires to think clearly about that the attacker can use what observation with the target model, the target model is a generative model or a discriminative model, and what detail prior information about dose the model the attacker has.

4) Considering with the prediction result, whether the adversary has the querying capability to get correspond prediction result with input data is one of the most important assumptions.

Understanding these aspects and developing an analysis structure serves a twofold purpose. First, it provides greater insight into previous researches, facilitating common ground comparison between different approaches. Second, it provides insights into the detailed design choices for MIA approaches which can contribute to the future research of membership inference attack against machine learning and the defense against the attack.

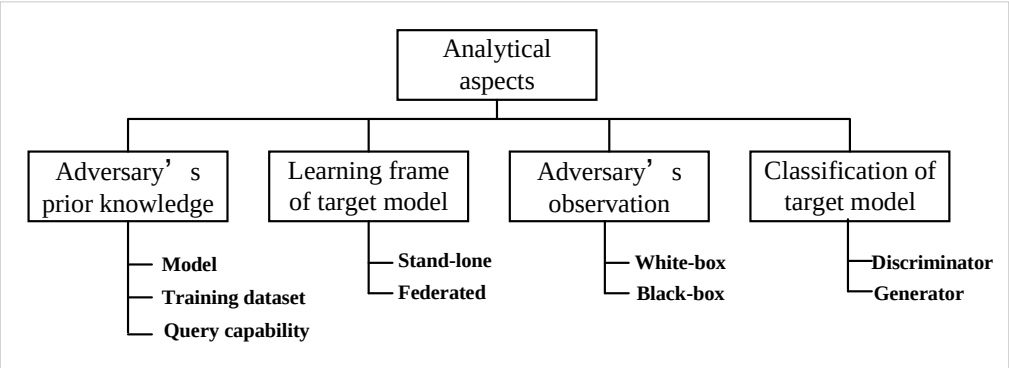


Figure 4: Analytical aspects of detail information

3.2 Analytical Framework

With all considerations mentioned above, we propose an analytical framework which includes adversary's prior knowledge, learning frame of target model, adversary's observation, classification of the target model, as shown in Figure 4. Then, we make definitions of these aspects and then study them in detail at the rest of this section. Lastly, we summary MIAs with our analytical framework in Section 3.7.

Adversary's prior knowledge. In an MIA, the more prior knowledge the adversary has, the stronger the adversary's capabilities are. On the contrary, the less prior knowledge the adversary has, the weaker the attacker's capabilities are. Several characteristics related to the adversary's capabilities. He *et al.*'s work [24] propose that the prior knowledge including three aspects: Knowledge of target model, knowledge of the training dataset, and the capability of the model querying. The prior knowledge will be comprehensive analyzed in Section 3.3.

Learning frame of the target model. The learning frame of the target model has two types, stand-alone learning frame, and federated learning frame. The adversary can different attack approaches with different learning frame.The collaborative learning will be discuss in Section 3.4 comparing with the stand-alone one.

Adversary's observation. In paper [45], they define that the adversary's observations of machine learning algorithms are what constitute the inputs for the MIA. The attack observation can be classified into the black-box and the white-box which will be discussed in Section 3.5.

Classification of the target model. There are two types of target model: Discriminative target model and the generative target model. Both of them suffer from MIA. The MIA against discriminative model and the generative model will be analyze in Section 3.6.

3.3 Adversary's Prior Knowledge

In this part, we consider the attacker's prior knowledge with completion coverage aspects, and then introduce a classification method of adversary's prior knowledge. In an MIA, the prior knowledge means the adversary's capabilities which have an impact on the attack results.

Previous works introduce several characteristics related to the adversary's capabilities. Salem *et al.* [53] studied three attacks with different prior knowledge consists of target model structure, training data distribution. He at el.'s work [24] classify the prior knowledge in three categories, *e.g.* target model, training dataset, the capability of model querying. This method covers all aspects related to MIA. Thus, we survey the previous works, by considering the prior knowledge into three aspects proposed by He *et al.* [24], *i.e.*, knowledge of target model, knowledge about the training dataset, the capability of model query. For clarity, we summarize the notations in Table 1.

Knowledge of Target Model (M). In this aspect, the adversary may obtain information about the target model, including the parameters, the structure, the type, machine learning as a service (MLaaS), and mode type, termed by M_1, M_2, M_3 and M_4 respectively.

M_1: Model parameters. Some researches [7, 22, 45] assumed that attacker knows some model parameters. The adversary can download the description of the model through MLaaS cloud systems [20, 42]. The method in [1] shows that an honest-but-curious server can partially recover participants' data points from the shared gradient updates. Paper [7] proposed an attack by which the attacker has access to the parameters of the generator.

M_2: Model structure. In paper [22,45,56], the adversary obtained knowledge of model structure.

M_3: MLaas platform. Several works [7,38,53,57,62] considered that the adversary can use the same MLaaS platform with target model.

M_4: Model type. Shokri *et al.* [57] and Hayes *et al.* [22] set the type of target model as one of the prior knowledge.

Table 1: Definition of prior knowledge's sub-classes

Knowledge Types	Symbol	Definition
Model	M_1	The parameters of the target model
	M_2	Can access or know the structure of the target model
	M_3	Use the same MLaaS platform with target mode
	M_4	The type of target model
Training Dataset	D_1	Know some properties about the target model's training dataset, such as the distribution, the size, or the value
	D_2	A dataset which includes model's training dataset
Query Ability	Q	Can query the machine learning model

Knowledge about the Training Dataset (D). Training data set is a set of examples used to initially fit the parameters (e.g. weights of connections) of the machine learning model. Each training example is represented by an array or vector, consists of pairs of an input vector and the corresponding output vector. There are many public dataset commonly used for machine learning model training, such as CIFAR-10,CIFAR-100, MNIST, Texas, Purchase-10, Purchase-100, Hospital, Location, News, and so on. The Knowledge about the training dataset means that the adversary has some information about the training dataset. the training dataset info includes the following two classes.

D_1: The attacker knows some property information about the target training dataset, such as the distribution, size or value. Long *et al.* [38] exploit datasets, which sampled from the same space as the terget training set but not containing the target record, to build the shadow model. Shokri *et al.* [57] say that they have some background knowledge about the target model's training dataset, but disjoint from the tarining dataset. Salem *et al.* [53] assume that the adversary has a dataset which comes from the same underlying distribution as the training data for the target model or perform model extraction to approximate the target model. Hayes *et al.* [22] give an assumption that the adversary knows the size of the training set, but not know how data-points are split into training and test sets.

D_2: The adversary obtains a dataset which includes model's training dataset.Nasr *et al.* [46] reveal that in a realistic setting, the probability distribution of data points and the probability distribution over the member of the training set are not directly and accurately available to the adversary. They assume a dataset known by the attacker which is the subset of the target training set. Hayes *et al.* [22] introduce a white-box attack in which the attacker has a dataset containing data-points used to train the target model.

The capability of Model Query (Q). This kind of prior knowledge means that the adversary whether can query the target model or not. In papers [7, 15, 22, 24, 37, 38, 52, 53, 57], the authors study that the adversary can query the learning model (Q).

3.4 The Frame of Target Model: Stand-Alone vs. Federated Learning

The learning frame of the target model has two major types, stand-alone learning one and federated learning one. It depends on whether all the training data is available in one place, or the training data is distributed among multiple parties [45]. The adversary has different attack approaches with different learning frames.

Stand-alone learning frame. In this setting, the target model is trained in one place, it means centralized training wherein all the training data is available in one place. Under the stand-alone learning frame, the adversary has two points of view to launch the MIA. First, the attacker can observe the process of the model updating. Second, an adversary can attack a final trained model. The latter method has been studied more than the former one, in previous works.

Federated learning frame. Comparing with the stand-alone learning frame, the federated learning frame has a distributed structure. The federated learning aims at training a machine learning algorithm on multiple local datasets contained in local participants. We illustrate the federated learning frame in Figure 5. In the federated learning frame, the central server orchestrates the different steps of the algorithm. First of all, the central server transmits the initial model to several distributed participants. Then, the participant uses local training datasets and optimizers (such as stochastic gradient descent optimizer) to train the local

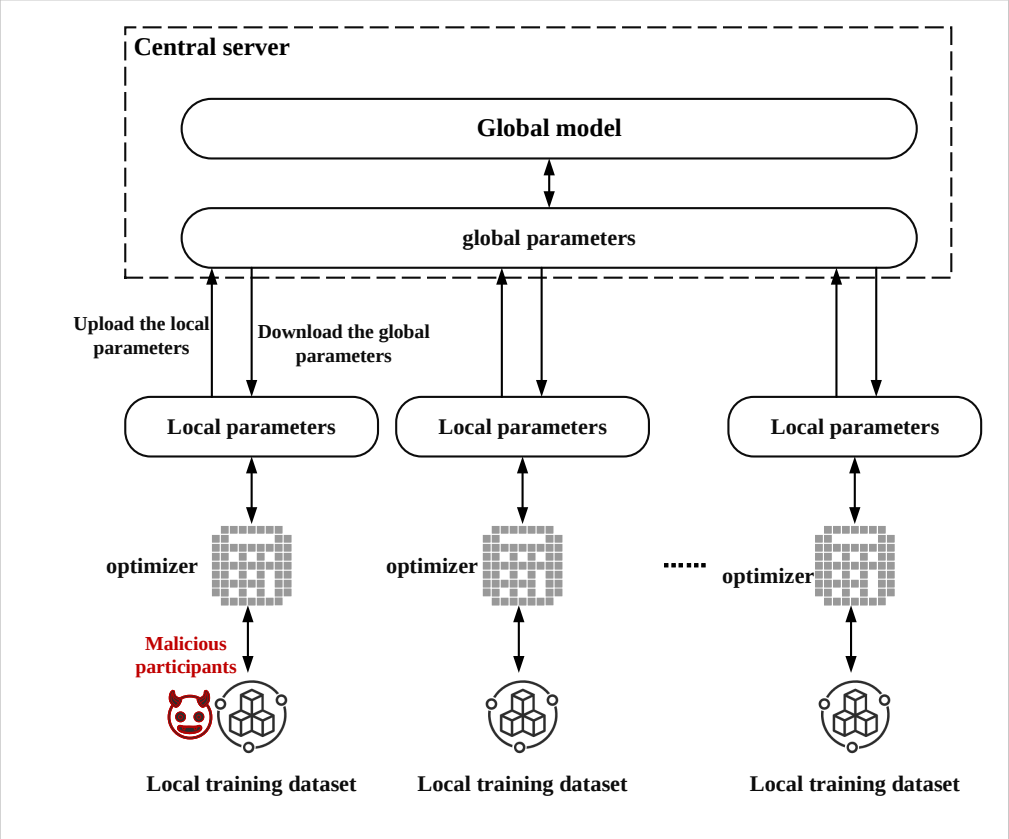


Figure 5: Federated learning frame

model. After that, participants upload their local parameters to the central server. The central server uses a specific method (such as computing average value) to transform these parameters into the global parameters. Finally, the central server generates a global model with global parameters. Federated learning constructs a global model using multiple rounds. In one round, it begins with downloading the global parameters from the central server and ends with uploading the local parameters to the central server. Collaborative training continues until the global model converges [45].

In papers [24, 41, 45], they propose an attack against federated learning, wherein the attacker is one of the participates who can observe the global parameters and craft his adversarial parameter updates to gain more information about the information of other participants' training dataset.

3.5 Observation: Black-Box vs. White-Box

White-box. A white box attack means that the attacker has access to the full model, notably its architecture and parameters. With such information, the adversary can reconstruct the target model and even the training dataset. Hayes *et al.*'s work [22] propose a white-box attack against the generative model. In their white-box scenario, the attacker relies on internal access to the target model instead of training an attack model. In paper [45], they introduce an extension of existing black-box membership to the white-box setting which uses the same attack on all of the activation functions of the model.

Black-box. In this setting, the adversary only has the capability of model querying but can neither access the model's parameters, nor the model structure. It means that the attacker can only query the target model by inputting data points to obtain output results. LOGAN [22] proposes a black-box attack with no auxiliary knowledge and a black-box attack with limited auxiliary knowledge. In ML-Leak [53], the authors present an attack whose adversary has black-box access to the target model, but the attacker not able to extract the membership status from the target model. Thus, the adversary trains a shadow model to mimic the behavior of the target model and relies on the shadow model to obtain the ground-truth membership to train the attack model. In paper [47], they introduce a black-box attack against the deep neural network (DNN) classifiers by crafting adversarial examples without knowledge of the classifier training data or model. In paper [57], the author defined a black-box attack in which the attacker used the given data record to query the target model in the black-box observation.

Among the two observations discussed above, the adversary under white-box setting is the most knowledgeable setting and the black-box observation has the least back-

ground knowledge. Therefore, white-box attacks are more powerful than black-box attacks. However, black-box attacks cannot be substituted by white-box attacks because the former is easier to apply in practice. For example, in a machine learning as a service (MLaaS) system, the attacker always has no knowledge about the target model's internal information, they do not know the model algorithm, have no knowledge about the model structure or the model parameter, but just has the capability to query the target model, so comparing with white-box MIAs, the black-box attacks are the most reasonable observation.

3.6 Classification of Target Model: Discriminative Model vs. Generative Model

Generally, machine learning models include discriminative models and Discriminative models. Both of them suffer from MIA and there are many previous works related to this aspect. In this section, we introduce the membership inference attack against discriminative models and generative models.

Discriminative models. Given the feature (x) of a data point and the corresponding label, discriminative models attempt to predict feature x by learning a discriminative function (x, y); The function takes in input x and outputs the most likely label y. It means that the discriminative models discriminate between different kinds of data points. However, discriminative models are not able to explain how the datapoints might have been generated [22]. Membership inference against discriminative deep learning models has attracted many studies [1, 3, 4, 27, 38, 41, 57, 66]. This kind of target model can provide confidence value about the data point which would help infer out the membership of the training dataset.

Generative models. Generative models describe how does the data generated by learning the joint probability distribution of $p(X, Y)$, which gives a score to the configuration determined together by pairs (x, y) [22]. Compared with the discriminative model, the membership inference attack against generative models has been less well-studied. As the generator cannot directly return the confidence value about the overfitting of data records, it's more challenging for the adversary in this scenario. With the generator model widely used in many applications, such as [2, 17, 18, 35, 40, 63], membership inference attacks against the generative model gained researchers' increasing attention. In the work [22], the authors use generative models to learn information about the target generative model, thus created a local copy of the target model for membership inference. In paper [7], the author proposed a taxonomy of membership inference attack against generative adversarial networks (GANs).

3.7 Summary

The summary of MIAs with analytical factors, which mentioned above, is provided by Table 2. In the existing work, researchers do more research on black-box membership inference attacks than white-box one. Among them, in the research of white-box, it is not necessary to know the conditions for query ability and information about training dataset; The attacker even only needs to know the M_1 condition to successfully obtain the membership information of the training dataset. For the MIA scenario of the discriminator, the attacker needs information related to the model and training dataset to assist in the attack. When one kind of the model or training dataset is missing, the query ability of the target model is needed to supplement the information. The above conditions are not necessary in the attack against the generator. The attacker can realize the MIAs on the generator through one kind of condition among the model attributes and training dataset properties. In the previous work, there are more attacks on stand-alone machine learning than attack against federated learning scenarios. The existing attacks against federated learning scenarios are usually white-box attacks, and at the same time, both information related to the model and training dataset are required to complete the MIAs. While the assumptions for stand-alone attacks will be more flexible.

4 Defenses

In this section, we discuss factors which influence the MIA. Then, we survey the defense mechanisms employed by existing privacy-preserving achievements and IMA defensive methods. Based on different implementation techniques, we classify the defense strategies into three categories: Generalization techniques, cryptography methods, adversarial method. In addition, we introduce prior work with these categories.

4.1 Factors Influence MIA

The factors influence MIA means that have on the advantage of adversaries who attempt to infer specific facts about the data used to train machine learning models [66]. Shokri *et al.* [57] show that overfitting is a sufficient condition for MI attack. The result in [38, 45] reveals that even well-generalized machine learning models might leak much information about their training data. Thus, it means that overfitting provides more information than necessary for MIA [38, 66]. Long *et al.*'s work [38] demonstrates that some training instances have unique impacts on the learning models, which will cause MI attacks. Shokri *et al.*'s work [57] finds that besides overfitting, the structure and type of the model also contribute to the vulnerable to MIA. Nasr *et al.* [45] show that model structure, gradients, and training size can also impact the learning model.

Table 2: Summary MIAs with our analytical aspects

Previous works	AO	Prior Knowledge				TMT	TF
		Model		TD	QA		
Long et al.'s [18]	B	M_3	D_1	Q	D		S
Shokri's [19]	B	M_3, M_4	D_1	Q	D		S
Nasr's-1 [20]	W	M_1, M_2	D_2	N	D		F
Nasr's-2 [20]	W	M_1, M_2	D_2	N	D		F
ML-Leaks-1 [22]	B	M_3	D_2	N	D		S
ML-Leaks-2 [22]	B	N	N	Q	D		S
ML-Leaks-3 [22]	B	N	N	Q	D		S
LOGAN-1 [23]	W	M_1, M_2, M_4	N	N	G		S
LOGAN-2 [23]	B	N	D_1, D_2	Q	G		S
LOGAN-3 [23]	B	N	D_1, D_2	Q	G		S
Gan-Leaks-1 [24]	B	M_3	N	Q	G		S
Gan-Leaks-2 [24]	B	N	D_1	N	G		S
Gan-Leaks-3 [24]	W	M_1	N	N	G		S
Nasr's et al.'s [26]	B	N	D_2	Q	D		S
Truex et al.'s [27]	B	M_3	N	Q	D		S
Melis et al.'s[33]	B	M_4	D_1	N	D		F
Long et al.'s[31]	B	N	D_1, D_2	Q	D		S

AO: attack observation, TD: training dataset, QA: query ability, TMT: target model type, TF: target frame; M_1, M_2, M_3, M_4, D_1, D_2, Q are the 7 sub-classes of adversary's prior knowledge, which defined in table I; N: not need; W: white-box; B: black-box; D: discriminator; G: generator. S: stand-alone; F: federated learning.

4.2 Generalization Techniques

As overfitting is an important reason why machine learning models leak information about their training datasets, generalization techniques such as dropout [53, 56, 58] can help degrade overfitting and strengthen privacy guarantees in neural networks [28] by randomly dropping out connections between neurons. While model stack [53] suitable for all machine learning models, independent of the classifier used to build them. The paper [57] uses standard regularization to overcome overfitting in machine learning.

4.3 Cryptography Methods

Homomorphic encryption. He *et al.* [24] use homomorphic encryption to encrypt the input in the collaborative learning scenario, so the sensitive information will not be leaked. A drawback of homomorphic encryption is inefficiency [24].

Differential privacy. Differential privacy has been regarded as a strong privacy standard [9–13]. The paper [61] presents a differentially private GANs model which includes a Gaussian noise layer in the discriminator if a generative adversarial network to make the output and the gradients differentially private with respect to the training data. The paper [4] uses the differentially-private stochastic gradient descent algorithm (DP-SGD) to prevent memorization. Salem *et al.*'s work [52] adds noise to the posterior for each queried sample and also adds noise sampled from a uniform distribution to the posteriors. The result shows that the method drops the attack accuracy to a certain degree. Especially, adding noise is hard work against a multi-sample reconstruction attack. In [67], the researcher introduces a data obfuscation function and applies it to the training data before feeding them to the model training task. By doing so, sensitive information about both the properties of individual samples and the statistical properties of a group of samples will be hidden. Jia *et al.* [32] propose to add noise to each confidence score vector predicted by the target model to turn the confidence score vector into an adversarial example, which can mislead the adversary's classifier to make random guessing at member and non-member.

4.4 Adversarial Method

In [46], Nasr *et al.* put forward a Min-Max Game which designs an adversarial training algorithm that minimizes the prediction loss of the model as well as the maximum gain of the inference attacks. This strategy, which can guarantee membership privacy acts also helped to generalize the target model. Jia *et al.* [32] proposes a method based on adversarial examples to mislead the attack model. There are many methods to find adversarial examples [5, 19, 34, 39, 43, 44, 48]. These adversarial methods may be exploited as defense strategies in the future.

5 Future Research Direction

The current MIA methods have the following problems: On the one hand, building MIAs requires many preconditions, such as: Information about data, model or query ability, which is unreasonable in actual scenarios; On the other hand, the current defense methods cannot have protective effects on various MIAs. Therefore, in the future, we can study MIA in realistic scenarios, approach the real world by reducing assumptions, and study effective general protection frameworks for MIA to solve these problems. Considering the current challenges and existing solutions, we expect that the research of MIA will be advanced in the following aspects.

Membership inference attack against federated learning frame would attract more attentions of researchers. Along with the widely application of machine learning, for obtaining better performance of model training, the learning frame gradually changed from stand-alone learning to the collaborative learning. Thus, there are much more sensitive data that would be used as the federated training dataset, such as location data, personal medical records, personal characteristic data, healthcare data. Such sensitive data would increase the adversary's interests. To study the attack methods under this scenario, and devise defensive strategies to mitigation these vulnerabilities has academic and application values.

Threats based on membership inference attack would be raised. In paper [22], the author indicated that membership inference attacks often act as a gateway to further attacks. The attacker can firstly infers whether the target data is a part of the training dataset, and then link up with other attacks, (e.g. profiling, property inference, which leak additional information about the victim, or other further attacks. Hence, the subsequent attacks after the launching of MIA would be studied in the future.

Another valuable topic for research is to find out a fully effective defensive methodology to cope with different attack approaches. The application of MI methods in security defense scenarios will draw more attention. Shokri *et al.*'s work [56] uses the membership inference method as defense mechanism.

6 Conclusions

The study of Membership Inference attack(MIA) against machine learning is quite young field. This research direction has attracted attention of scholars and offers a number of opportunities for future exploration. For researchers just entering MI attacks and defenses against machine learning, we provided an in-depth introduction to this research field in its current state. For active researchers in the field, this paper not only provide a structured and comprehensive survey, but also as fundamental knowledge for the future researches in this area.

In this artical, we summarize the year-wise road map of MIAs against machine learning. and then, we construct a comprehensive framework to analyze the existing MIAs against machine learning systems, classifies the adversary's prior knowledge into seven sub-classes, overviews the factors that influence the attacks; Next we analyze the prior works with our framework, and give out a systematic comparison in Section 3.7. Further More, we characterizes existing defense mechanisms for MIA against machine learning in to three categories. Lastly, we give out the future research direction in this field.

Acknowledgments

This study was supported by a grant from the National High Technology Research and Development Program of China(863 Program)(No.2009AA01Z435).

References

[1] Y. Aono, T. Hayashi, L. Wang, S. Moriai, *et al.*, "Privacy-preserving deep learning: Revisited and enhanced," in *International Conference on Applications and Techniques in Information Security*, pp. 100–110, 2017.

[2] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 224–232, 2017.

[3] M. Backes, P. Berrang, M. Humbert, and P. Manoharan, "Membership privacy in microrna-based studies," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, pp. 319–330, 2016.

[4] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D Song, "The secret sharer: Measuring unintended neural network memorization & extracting secrets," *Machine Learning*, 2018. arXiv:1802.08232.

[5] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy (SP'17)*, pp. 39–57, 2017.

[6] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *Ieee Access*, vol. 5, pp. 8869–8879, 2017.

[7] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "GAN-leaks: A taxonomy of membership inference attacks against gans," *Machine Learning*, 2019. arXiv:1909.03935.

[8] M. Y. Chen, C. C. Yang, and M. S. Hwang, "Privacy protection data access control," *International Journal Network Security*, vol. 15, no. 6), pp. 411–419, 2013.

[9] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*, pp. 1–19, 2008.

[10] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 486–503, 2006.

[11] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pp. 371–380, 2009.

[12] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[13] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60, 2010.

[14] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature medicine*, vol. 25, no. 1, pp. 24–29, 2019.

[15] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333, 2015.

[16] T. T. Gao, H. Li, and S. L. Yin, "Adaptive convolutional neural network-based information fusion for facial expression recognition," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 17-23, 2021.

[17] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *Machine Learning*, 2014. arXiv:1412.6572.

[20] Google, *AI Platform*, 2020. (`https://cloud.google.com/ai-platform`)

[21] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *Computation and Language*, 2014. arXiv:1412.5567.

[22] J. Hayes, L. Melis, G. Danezis, and E. D. Cristofaro, "LOGAN: Evaluating information leakage of generative models using generative adversarial networks," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 1, 2017.

[23] Y. He, S. Rahimian, B. Schiele, and M. Fritz, "Segmentations-leak: Membership inference attacks and defenses in semantic image segmentation," *Computer Vision and Pattern Recognition*, 2019. arXiv:1912.09685.

[24] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 148–162, 2019.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[26] B. Hilprecht, M. Härterich, and D. Bernau, "Monte carlo and reconstruction membership inference attacks against generative models," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, pp. 232–249, 2019.

[27] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, pp. 603–618, 2017.

[28] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density snp genotyping microarrays," *PLoS Genetics*, vol. 4, no. 8, 2008.

[29] M. S. Hwang, E. F. Cahyadi, S. F. Chiou, and C. Y. Yang, "Reviews and analyses the privacy-protection system for multi-server," in *Journal of Physics: Conference Series*, vol. 1237, pp. 022091, 2019.

[30] M. S. Hwang and I. C. Lin, "Introduction to information and network security (in chinese)," *Mc Grew Hill. In Taiwan*, 4, 2011.

[31] M. S. Hwang, C. H. Wei, and C. Y. Lee, "Privacy and security requirements for RFID applications," *Journal of Computers*, vol. 20, no. 3, pp. 55–60, 2009.

[32] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, pp. 259–274, 2019.

[33] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, vol. 2, 2015. (`https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf`)

[34] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *Computer Vision and Pattern Recognition*, 2016. arXiv:1607.02533.

[35] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European Conference on Computer Vision*, pp. 702–716, 2016.

[36] Z. Li and Y. Zhang, "Label-leaks: Membership inference attack with label," *Machine Learning*, 2020. arXiv:2007.15528.

[37] Y. Long, V. Bindschaedler, and C. A. Gunter, "Towards measuring membership privacy," *Cryptography and Security*, 2017. arXiv:1712.09136.

[38] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, "Understanding membership inferences on well-generalized learning models," *Computer Science*, 2018. arXiv:1802.04889.

[39] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *Machine Learning*, 2017. arXiv:1706.06083.

[40] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," *Sound*, 2016. arXiv:1612.07837.

[41] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *IEEE Symposium on Security and Privacy (SP'19)*, pp. 691–706, 2019.

[42] Microsoft, *Microsoft Azure Machine Learning*, 2020. (`https://azure.microsoft.com/en-us/services/machine-learning/`)

[43] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1765–1773, 2017.

[44] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, 2016.

[45] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *IEEE Symposium on Security and Privacy (SP'19)*, pp. 739–753, 2019.

[46] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, pp. 634–646, 2018.

[47] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of ACM on Asia Conference on Computer and Communications Security*, pp. 506–519, 2017.

[48] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy (EuroS&P'16)*, pp. 372–387, 2016.

[49] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro, "Knock knock, who's there? Membership inference on aggregate location data," *Proceedings of the 25th Network and Distributed System Security Symposium*, 2017. arXiv:1708.06145.

[50] S. Rahimian, T. Orekondy, and M. Fritz, "Differential privacy defenses and sampling attacks for membership inference," in *PriML Workshop (PriML'19)*, vol. 13, 2019. (`https://priml-workshop.github.io/priml2019/papers/PriML2019_paper_47.pdf`)

[51] A. Sablayrolles, M. Douze, Y. Ollivier, C. Schmid, and H. Jégou, "White-box vs black-box: Bayes optimal strategies for membership inference," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 5558-5567, 2019.

[52] A. Salem, A. Bhattacharya, M. Backes, M. Fritz, and Y. Zhang, "Updates-leak: Data set inference and reconstruction attacks in online learning," *Cryptography and Security*, 2019. arXiv:1904.01067.

[53] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *Computer Science*, 2018. arXiv:1806.01246.

[54] S. Sankararaman, G. Obozinski, M. I. Jordan, and E. Halperin, "Genomic privacy and limits of individual detection in a pool," *Nature Genetics*, vol. 41, no. 9, pp. 965, 2009.

[55] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Over-the-air membership inference attacks as privacy threats for deep learning-based wireless signal classifiers," in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, pp. 61–66, 2020.

[56] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1310–1321, 2015.

[57] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE Symposium on Security and Privacy (SP'17)*, pp. 3–18, 2017.

[58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[59] J. X. Tong, H. Li, and S. L. Yin, "Research on face recognition method based on deep neural network," *International Journal of Electronics and Information Engineering*, vol. 12, no. 4, pp. 182–188, 2020.

[60] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *The 25th USENIX Security Symposium*, pp. 601–618, 2016.

[61] A. Triastcyn and B. Faltings, *Generating Differentially Private Datasets Using Gans*, 2018. (`https://openreview.net/pdf?id=rJv4XWZA-`)

[62] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service,"

IEEE Transactions on Services Computing, 2019. DOI:10.1109/TSC.2019.2897554.

[63] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *Sound*, 2016. arXiv:1609.03499.

[64] C. H. Wei, M. S. Hwang, and A. Y. H. Chin, "A secure privacy and authentication protocol for passive RFID tags," *International Journal of Mobile Communications*, vol. 15, no. 3, pp. 266–277, 2017.

[65] C. H. Wei, M. S. Hwang, and A. Y. H. Chin, "Security analysis of an enhanced mobile agent device for RFID privacy protection," *IETE Technical Review*, vol. 32, no. 3, pp. 183–187, 2015.

[66] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *IEEE 31st Computer Security Foundations Symposium (CSF'18)*, pp. 268–282, 2018.

[67] T. Zhang, Z. He, and R. B. Lee, "Privacy-preserving machine learning through data obfuscation," *Cryptography and Security*, 2018. arXiv:1807.01860.

[68] G. Zhang, A. Zhang, and P. Zhao, "Locmia: Membership inference attacks against aggregated location data," *IEEE Internet of Things Journal*, pp. 1-1, 2020. DOI: 10.1109/JIOT.2020.3001172.

Biography

Yang Bai is the Ph.D candidate of the University of Electronic Science and Technology of China (UESTC), China. Her research interest include machine learning, cloud computing, and information security.

Ting Chen received the Ph.D degree from the University of Electronic Science and Technology of China (UESTC), China, 2013. Now he is a professor with UESTC. His research interest include on blockchain, smart contract, program analysis, and information security.

Mingyu Fan received her Ph.d in Southwest Jiaotong University, and worked as a post-doc in Tsinghua University. Now she is a professor at the University of Electronic Science and Technology of China(UESTC), China. Her research interest is mainly in the area of communication engineering, computer science, and information security.

A BP Neural Network-oriented Henon Hyperchaotic System for Image Encryption

Desheng Liu, Fuqiang Wang, and Hui Wang
(Corresponding author: Hui Wang)

School of information and electronic technology, Jiamusi University
Jiamusi 154007, China
Email: wh4922866@163.com

(Received Jan. 9, 2020; Revised and Accepted Aug. 21, 2020; First Online June 10, 2021)

Abstract

The traditional image encryption method can not resist the attack of statistical characteristics. Therefore, the low dimensional chaotic ciphertext is easily cracked. Therefore, this paper proposes a BP neural network-oriented Henon Hyperchaotic System for image encryption. BP neural network has the ability to realize any complex nonlinear mapping, so the chaotic sequence is obtained through BP neural network in the process of encryption, the complexity of the algorithm will be increased. Then, the target key and the parameters related to the image pixel are introduced to redesign the S-box and wheel key to solve the disadvantage that the S-box and wheel key is fixed for each round. Finally, two more rounds of AES (Advanced Encryption Standard) circular encryption are performed to increase the complexity of the encryption process. Experimental simulation shows that this algorithm can effectively improve the security of the proposed algorithm.

Keywords: AES; BP Neural Network; Henon Hyperchaotic System; Image Encryption

1 Introduction

In recent years, with the rapid development of Internet technology, the using of multimedia for information exchange has become an important way for people to communicate. Images have been widely used because of the rich information and vivid expression [5,13]. But the security of images has become a concern problem. Therefore, the efficient and secure encryption of images has aroused people's great attention. Up to now, many ideal image encryption algorithms have been proposed, but these algorithms still have some shortcomings [8,9]. For example, the encryption algorithm based on DNA rules has very limited coding addition and subtraction operation rules and low security [6]. The encryption method based on one encryption at a time requires that the size of key is same as the plaintext, so the feasibility is not high. The encryption method based on Bit rules requires lots of computation and it has low encryption efficiency, so it is not suitable for actual image encryption. In the perceptron model-based encryption method, there is no connection between the equivalent key flow and the ciphertext image. The encryption method based on reversible cellular automata adopts multiple cycles and iterations for image pixels with very low encryption efficiency. And the key has great limitations and is vulnerable to exhaustive attacks [11, 14].

Chaotic systems are often used in image encryption due to the sensitivity to initial values, pseudo-randomicity and aperiodicity. Low-dimensional chaotic system has the disadvantages with simple structure, smaller key space, poor chaotic sequence randomness. it cannot resist common attack effectively. However, the hyperchaotic system has more than two positive Lyaponuv indices, which have very good pseudo-randomicity and large key space, so it has higher chaotic characteristics [15, 16]. The nonlinear dynamic behavior is more complicated and difficult to predict, which will undoubtedly increase the complexity of the algorithm and enhance the security in the encryption process.

Neural networks have very complex nonlinear dynamic behaviors and can process signals in real time [2, 4]. Ryosuke [18] first introduced biological neurons into neural networks to predict nonlinear systems with complex chaotic dynamics. In the literature, Hopfield neural network was introduced into image encryption, and chaos control parameters were introduced into the encryption process to design scrambled diffusion structure. In [3], it introduced cellular neural network and designed an image encryption algorithm to solve the problem that stream cipher was insensitive to the change of plaintext. Through these excellent algorithms, it can be understood that the nonlinear sequences obtained from chaotic sequences trained by neural network are more random and disordered.

In view of the shortcomings of the above image encryption methods, this paper proposes an image encryption al-

gorithm based on the improved Henon hyperchaotic system combined with AES and BP neural network. The high-dimensional Henon hyperchaotic system has more than two positive Lyaponuv indices, very complex nonlinear dynamic behavior and large key space, which can make up the defect of small key space in AES encryption algorithm [1, 19, 20]. BP neural network has the ability to realize any complex nonlinear mapping, so the nonlinear sequence obtained from chaotic sequence through BP neural network will be more randomness and disorder. That will increase the complexity of the algorithm in the encryption process. First, the four-dimensional Henon hyperchaotic system is used to generate hyperchaotic sequences. By introducing the average value of the plaintext image pixel as the parameter, the chaotic sequence is intercepted as the training sample of BP neural network, and the trained nonlinear chaotic sequence is used as the target key of AES encryption algorithm. Next, the plaintext image is processed by XOR with the target key. It can prevent other phases that do not need the key, and can be calculated in reverse in the case of unknown key, which can effectively enhance the security of the algorithm. Then, the target key and the parameters related to the image pixel are introduced to redesign the S-box and wheel key to solve the disadvantage that the S-box and wheel key are fixed for each round. Finally, two more rounds of AES circular encryption are performed to increase the complexity of the encryption process. Experimental simulation shows that the algorithm can effectively combine the advantages of them, so that the effectiveness and security of proposed method are improved comprehensively.

2 Proposed Encryption Algorithm

In this paper, the hyperchaotic sequence is first generated by the four-dimensional Henon hyperchaotic system. The average value of the plaintext image pixel is introduced as the parameter to intercept the chaos, and the sequence is used as the training sample of BP neural network. After the training, the nonlinear chaotic sequence is obtained as the target key of AES encryption algorithm. Finally, the image is encrypted by the double round AES algorithm.

2.1 Four-Dimensional Henon Hyperchaotic System

Chaos is the discontinuous, irregular and unstable motion behavior in nonlinear dynamic system. The first chaotic system was discovered by Lorenz in 1963. Then new chaotic systems have been proposed ever since. In 1976, Henon studied two-dimensional functions and proposed the Henon hybrid system, which was a more complex system.

In 2002, Richter [10] defined a generalized Henon chaotic system, the expression was shown in Formula (1):

$$\begin{cases} x_1(k+1) = a - x_{\omega-1}(k)^2 - bx_\omega(k) \\ x_i(k+1) = x_{i-1}(k) \end{cases} \tag{1}$$

Where, ω represents the dimension. When $\omega > 2$, the system will be hyperchaotic. a and b are the control parameters, $i = 2, 3, \cdots, \omega$. This paper studies the four-dimensional Henon hyperchaotic system, *i.e.* $\omega = 5$, as shown in Equation (2).

$$\begin{cases} x_1(k+1) = a - x_3(k)^2 - bx_4(k) \\ x_2(k+1) = x_1(k) \\ x_3(k+1) = x_2(k) \\ x_4(k+1) = x_3(k) \end{cases} \tag{2}$$

Simulation experiments prove that hyperchaos will occur when chaos control parameters $a = 1.76, b = 0.1$ and initial value $x_j(1) = 1, j = 1, 2, \cdots, 4$.

2.2 BP Neural Network

Rumelhart [21] proposed BP (Back Propagation) neural network, which was a kind of multi-layer feed-forward neural network. A BP neural network is composed of neurons and the topology of the network. The neuron model is shown in Figure 1. The neuron has n inputs, *i.e.* $a_i(i = 1, 2, \cdots, n)$. The topology of network is the interconnection structure between neurons. BP neural network is composed of input layer, hidden layer and input layer. Hidden layer may have one or more layers. The three-layer BP network structure is shown in Figure 2. BP neural network has very high fault-tolerant capacity, even if the local neuron is destroyed, the whole training result will not be affected. Moreover, the hidden layer in BP neural network can approach a nonlinear function with arbitrary accuracy under the premise of enough hidden nodes, so BP neural network can establish a highly complex nonlinear relationship between input and output.

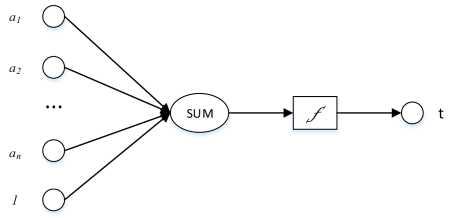


Figure 1: Neuron model

2.3 AES Encryption Algorithm

In this paper, XOR operation is carried out on the plaintext image and the target key. Then the double AES algorithm is used to encrypt data two times. Each round of AES is divided into four steps:

1) S-Box transformation. The first four bits of binary are taken as row coordinates, and the last four bits of binary are taken as column coordinates;

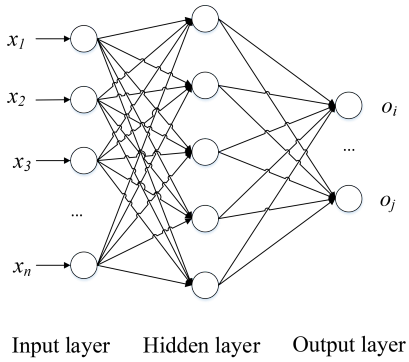


Figure 2: BP three-layer topology structure

2) Row shift operation. It shifts n bytes to the right in the matrix;

3) Column obfuscation operation. A set of matrices and ciphertext are replaced by new columns obtained by XOR transformation;

4) Round key addition. Ciphertext and key are processed by XOR.

The encryption process of AES algorithm is shown in Figure 3.

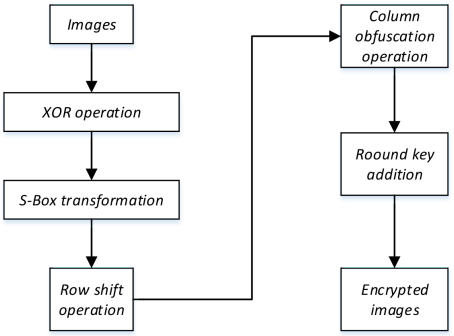


Figure 3: AES algorithm

2.4 Encryption and Decryption Process

2.4.1 Encryption Principle

Gray image are usually generated by weighted sum. In order to facilitate algorithm discussion, let A represent M × N grayscale image, where avg is the average pixel. The encryption algorithm process is shown in Figure 4.

The following is the detailed process.

Step 1. The plaintext image A is transformed into the matrix P with size $M \times N$ in line priority order, the range is [0,255].

Step 2. We obtain the sum of all elements in matrix P. The average pixel value of the image is obtained by Formula (3).

$$avg = \text{sum}\,/M \times N \tag{3}$$

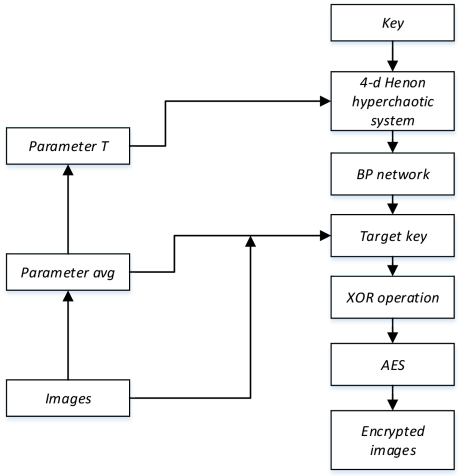


Figure 4: Encryption process

Step 3. Setting four initial values X, Y, Z, W. To eliminate the adverse effects caused by transient effect in 4-d Henon hyperchaotic system, let the four-dimensional Henon hyperchaotic system iterate the T wheel, in which T is obtained by Formula (4). It starts with time $T + 1$ and operates M × N times. This can generate four chaotic sequence X, Y, Z, W with a length M × N.

$$T = 1000 + \text{mod}\left(\text{avg} \times 10^{10}, 10 \times (M \times N)\right) \tag{4}$$

Step 4. Three-layer BP neural network is used to train the four chaotic sequences X, Y, Z, and W. The training parameters of neural network are set as follows: the weight between the input layer and the hidden layer is ; the weight between the hidden layer and the output layer is $\omega_i = 1 (i = 1, 2, \cdots, M \times N)$. The excitation function adopts the Laguerre function as shown in Equation (5).

$$f(x) = e^x \frac{d^n\left(x^n e^{-x}\right)}{dx^n}, 0 \le x < +\infty \tag{5}$$

The recursive method is adopted to take the new sequence learned by training as the input of the data network. The new four chaotic sequences X, Y, Z, and W are obtained by repeating this operation for 1000 times. The value obtained after rounding the chaotic sequence data and dividing by 256 is taken as the target key, so the value range of the target key is [0,255].

Step 5. It transforms the chaotic sequence X_1 to a matrix with $M \times N$, and obtains the image A_1 after one XOR transformation with the matrix P.

Step 6. S-box design. The data from avg to $avg + 256$ of chaotic sequence Y_1 are intercepted into the target key, which is represented by sequence $K = \{k_1, k_2, \cdots, k_{256}\}$. Sequence H is generated after that the elements in sequence K are sorted from small

to large. The row number and column number of S-Box are designed to be the first four bits and the last four bits of each element value in ciphertext A_1 respectively. Then the element value is replaced with the value of the corresponding position of S-Box, and the matrix A_2 is calculated.

Step 7. It carries out row shift transformation and scrambles the ciphertext matrix by shifting n bits to the right of the n-th row to obtain matrix A_3.

Step 8. Column confounding. Every 4 elements in matrix sequentially use matrix A_3 [02,03,01,01;01,02,03,01;01,01,02;03;03,01,01,02] to perform the XOR operation in the way of row priority.

Step 9. Performing the round key addition operation. It transforms the chaotic sequence Z_1 into the matrix with $M \times N$ and performs the XOR transformation with the ciphertext matrix.

Step 10. Repeat Steps 6 9 for another round of AES algorithm encryption. In here, the S-Box of Step 6 is generated by the sequence values from the avg +257 to avg +512 of chaotic sequence Y_1. The chaotic sequence of round key encryption in Step 9 is chaotic sequence W_1. Two rounds of AES algorithm encryption are completed to obtain the final ciphertext image.

2.4.2 Decryption Principle

Decryption is the reverse process of encryption. Firstly, the key and corresponding parameters are substituted into the four-dimensional Henon hyperchaotic system and BP neural network to generate the target key. Then, the target key and ciphertext image are reverse-processed by AES algorithm, and the plaintext image is finally obtained. The decryption process is shown as Figure 5.

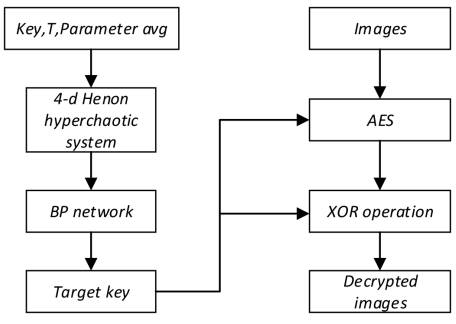


Figure 5: Decryption process

3 Experiments and Analysis

In this paper, the simulation experiment is carried out under the environment of Matlab 2016a. For the convenience of verification, Lena, Cameraman and Baboon gray scale images with the size of 256×256 are selected as the plaintext images in the this experiment. The encryption scheme in this paper is also suitable for other sizes. The initial key is $x_i = 1 (i = 1, 2, 3, 4)$. The plaintext image and ciphertext image of Lena, cameraman, Baboon shown in Figure 6 are obtained through the proposed encryption scheme. Obviously, our proposed encryption method works better.

Figure 6: The first row is original image; The second row is encrypted image

3.1 Statistical Analysis of the Proposed Algorithm

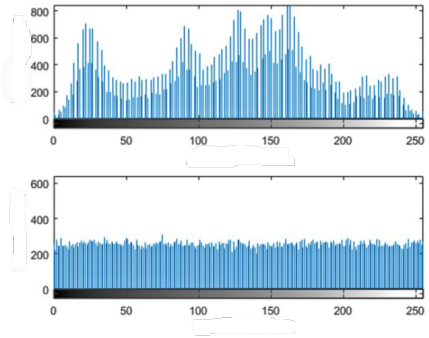
Histogram analysis. The histogram of grayscale image represents the frequency of each grayscale pixel, the abscissa is the grayscale, and the ordinate is the frequency of the grayscale, which can effectively reflect the distribution of image grayscale. The histogram distribution of secure ciphertext images is very uniform, which can effectively prevent the decoder from extracting valuable information from the histogram of ciphertext images. The gray histograms of plaintext and ciphertext image of Lena, Cameraman and Baboon are shown in Figure 7. Obviously, the distribution of gray histogram of plaintext image is not uniform, and that of ciphertext image is very uniform. Therefore, the ciphertext image can effectively store the distribution characteristics of the plaintext image. The encryption effect is better, and the security is very high.

The variance of histogram is to quantitatively analyze the uniformity of histogram. If the variance is smaller, the histogram is more uniform. Variance calculation formula is as follows:

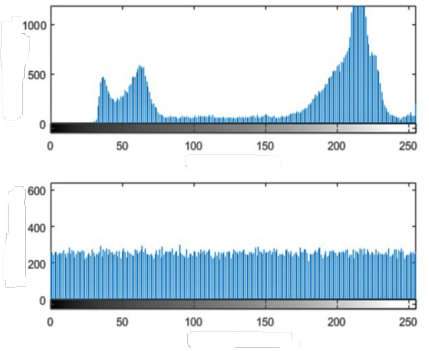
$$\mathrm{var}(Z) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} 0.5 \left(z_i - z_j\right)^2 \tag{6}$$

Where Z is the histogram pixel value vector, $n = 1, 2, \cdots, 256$.

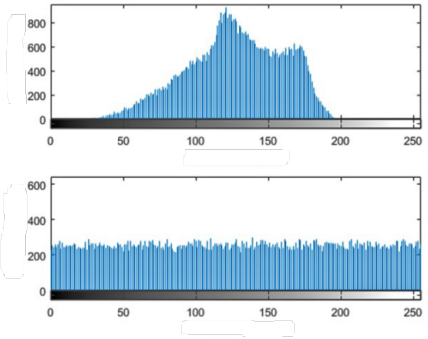
Histogram variance results of plaintext and ciphertext images of Lena, Camerman and Baboon are shown in Table 1. Obviously, the histogram variance of ciphertext images is much smaller than that of plaintext images, indicating that the histogram of ciphertext is very uniform and achieves better encryption effect.



(a) Histogram of plaintext and ciphertext image Lena.



(b) Histogram of plaintext and ciphertext image Cameraman.



(c) Histogram of plaintext and ciphertext image Baboon.

Figure 7: Histogram distribution

Table 1: Histogram variance of plaintext and ciphertext images

Image	Plaintext	Ciphertext
Lena	40032	251
Camerman	92838	256
Baboon	44695	253

Algorithmic statistical analysis. A random set of adjacent pixel values is extracted in plaintext and ciphertext to test the correlation (including vertical, horizontal, and diagonal directions) and calculate the correlation coefficient. The correlation coefficient can be calculated according to Equations (7)-(10).

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{7}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})^2 \tag{8}$$

$$\mathrm{Conv}(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y}) \tag{9}$$

$$r = \frac{\mathrm{Conv}(x,y)}{\sqrt{D(x)} \times \sqrt{D(y)}} \tag{10}$$

Where $\mathcal{X}$ and y represent the pixel values of two adjacent pixels. The relationship between the pixel values of adjacent horizontal points of Lena plaintext and ciphertext image is shown in Figure 8. It can be seen that the correlation between the pixel values of adjacent points in the plaintext image is very high, but there is no correlation between the pixel values of adjacent points in the ciphertext image.

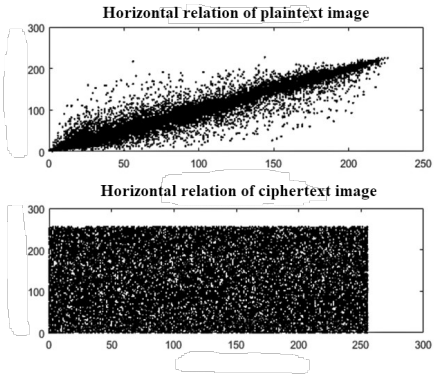


Figure 8: Correlation of adjacent pixels

A set of adjacent pixel values are randomly selected from Lena, Cameraman and Baboon plaintext and ciphertext image to test their correlation. Table 2 shows the correlation coefficients and mean value before and after encryption. Table 3 shows the comparison results of the average ciphertext correlation coefficient with different methods (including GBS [12], FDC [7], SCKG [22] and ECCHE [17]). Thus, the adjacent pixels of ciphertext image have almost no relationship. It shows that the proposed algorithm in this paper has achieved a better encryption effect and has a strong ability to resist statistical analysis.

3.2 Against Differential Attack

The resistance to differential attack is mainly analyzed from the sensitivity of the key and the sensitivity of the plaintext.

Sensitivity analysis of key. The sensitivity of the key means that once the decryption key changes very slightly,

Table 2: The correlation coefficient between the adjacent pixels of the plaintext image and the encrypted image

Image	Horizontal direction	Vertical direction	Diagonal direction
Lena plaintext	0.9625	0.9352	0.9232
Cameraman plaintext	0.9561	0.9618	0.9271
Baboon plaintext	0.7127	0.6535	0.6211
Lena ciphertext	-0.0013	-0.0411	0.0022
Cameraman ciphertext	-0.0028	-0.0112	0.0018
Baboon ciphertext	0.0023	0.0035	0.0034
Mean value	0.0022	0.0184	-0.0025

Table 3: Comparison results

Image	Horizontal direction	Vertical direction	Diagonal direction
GBS	0.0091	0.0193	0.0012
FDC	0.0073	0.0165	0.0007
SCKG	0.0045	0.0157	-0.0018
ECCHE	0.0029	0.0132	-0.0021
Proposed method	0.0022	0.0184	-0.0025

the decryption image will change significantly. Through experiment, the correct decryption method is adopted to decrypt Lena ciphertext image to obtain decrypted image A, as shown in Figure 9(a).

The key is changed from $x_1 = 1$ to $x_1 = 1 + 10^{-10}$, the decryption image is shown in Figure 9(b). The key is changed from $x_2 = 1$ to $x_2 = 1 + 10^{-10}$, the decryption image is shown in Figure 9(c). The key is changed from $x_3 = 1$ to $x_3 = 1 + 10^{-10}$, the decryption image is shown in Figure 9(d). The key is changed from $x_4 = 1$ to $x_4 = 1 + 10^{-10}$, the decryption image is shown in Figure 9(e). Obviously, if the initial key only has a small change of 10^{-10}, the image will not be decrypted, which indicates that the sensitivity of the key is very high.

Sensitivity analysis of the plaintext. The plaintext sensitivity indicates that if there are very small changes in the image of the plaintext image, the ciphertext will also have a large change. If the plaintext sensitivity is higher, the algorithm's ability to resist differential attack is stronger. This is because the image encryption process can be deduced from the small changes in the plaintext image. Therefore, in this paper, plaintext features are introduced in the encryption process, that is, the average value of the plaintext image pixel is taken as the encryption parameter. Once the plaintext changes slightly, the ciphertext will also change significantly.

The sensitivity analysis of the plaintext mainly refers to the Number of Pixels Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). Two encrypted images G_1 and G_2 , where G_1 represents the encrypted image of the original plaintext image. G_2 is the encrypted image after changing the pixel value of the original plaintext image at position (i.j). Define a matrix P with $M \times N$ size. When $G_1(i,j) = G_2(i,j)$, the matrix $P(i,j) = 0$; Otherwise, the matrix $P(i,j) = 1$. The rows and columns of the encrypted image are represented by M and N, respectively.

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\%$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|M_1(i,j) - M_2(i,j)|}{255} \times 100\%$$

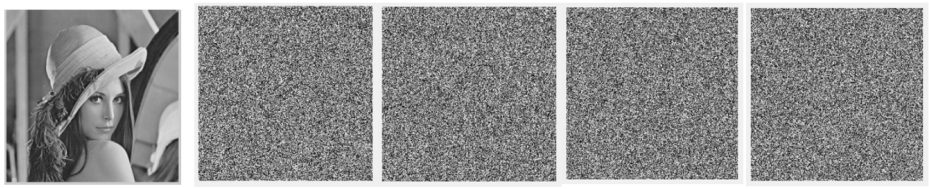
Table 4 is the NPCP and UACI comparison result (Take Lena as an example). It can be seen that this proposed algorithm is very sensitive to plaintext and can resist differential attacks effectively.

Table 4: NPCP and UACI comparison results

Index	GBS	FDC	SCKG	ECCHE	Our
NPCP/%	99.37	99.55	99.61	99.68	99.72
UACI/%	27.38	28.73	29.84	32.54	34.26

3.3 Key Space Analysis

When the key space is large enough, the encryption algorithm's resistance to exhaustive attacks must be very strong, and it can improve the security of encryption. The initial key used in this paper is the double precision type, so the valid data for the initial key can reach to 15-16 bits. According to the four initial keys entered by the four-dimensional Henon hyperchaotic system, the key space can reach at least 10^{60}, which is larger than the space 2^{128} of the traditional AES algorithm. Therefore, the encryption algorithm in this paper has a large key space and



(a) (b) (c) (d) (e)

Figure 9: (a) correct key; $(b)x_1 = 1 + 10^{-10}$; (c) $x_2 = 1 + 10^{-10}$; (d) $x_3 = 1 + 10^{-10}$; (e) $x_4 = 1 + 10^{-10}$

strong ability to resist exhaustive attacks. Table 5 is the encryption time comparison.

Table 5: Encryption time/s comparison results

Image	A	B	C	D	Our
Lena	4.839	4.256	3.717	2.884	1.752
Cameraman	4.325	3.684	3.542	2.189	1.395
Baboon	4.769	4.062	3.554	2.374	1.582

4 Conclusion

Aiming at the shortcomings of traditional AES encryption algorithm, such as small key space and fixed size, this paper proposes an improved image encryption algorithm combining Henon hyperchaos system and BP neural network. The four-dimensional Henon hyperchaotic system combined with BP neural network produces the target key of AES encryption algorithm, which greatly improves the randomness of the key. Also it introduces the relevant characteristics of plaintext image as parameters, which can effectively resist the plaintext attack. Experimental results show that the proposed algorithm is very effective in image encryption.

Acknowledgments

This research was supported by Basic scientific research project of Heilongjiang Provincial University (No. 2018-KYYWF-0942).

References

[1] A. B. Abugharsa, A. S. B. H. Basari, H. Almangush, "A new image encryption approach using the integration of a shifting technique and the AES algorithm," *International Journal of Computer Applications*, vol. 42, no. 9, pp. 36-45, 2012.

[2] E. N. Ganesh, "Study of VoIP network delay using neural networks," *International Journal of Electronics and Information Engineering*, vol. 12, no. 2, pp. 83–91, 2020.

[3] L. U. Huibin, W. Lijia, "Color image encryption algorithm of chaotic based on the hopfield network," *Journal of Jilin University*, vol. 32, no. 2, pp. 131-137, 2014.

[4] M. S. Hwang, C. C. Chang, K. F. Hwang, "Digital watermarking of images using neural networks", *Journal of Electronic Imaging*, vol. 9, no. 4, pp. 548–555, Jan. 2000.

[5] T. Lin, H. Li and S. Yin, "Modified pyramid dual tree direction filter-based image de-noising via curvature scale and non-local mean multi-grade remnant multi-grade remnant filter," *International Journal of Communication Systems*, vol. 31, no. 16, Nov. 10, 2018.

[6] H. Liu, X. Wang, A. kadir, "Image encryption using DNA complementary rule and chaotic maps," *Applied Soft Computing*, vol. 12, no. 5, pp. 1457–1466, 2012.

[7] Z. Liu, T. Xia, "Novel two dimensional fractional-order discrete chaotic map and its application to image encryption," *Applied Computing & Informatics*, vol. 14, no. 2, pp. 177-185, 2018.

[8] S. Mozaffari, "Parallel image encryption with bit-plane decomposition and genetic algorithm," *Multimedia Tools & Applications*, vol. 77, no. 10, pp. 1-21, 2018.

[9] B. Ramalingam, D. Ravichandran, A. A. Annadurai, *et al.*, "Chaos triggered image encryption-a reconfigurable security solution," *Multimedia Tools & Applications*, vol. 10, pp. 1-24, 2017.

[10] H. Richter, "The generalized hnon maps: Examples for higher-dimensional chaos," *International Journal of Bifurcation and Chaos*, vol. 12, no. 06, pp. 1371-1384, 2002.

[11] A. Sarkar, J. K. Mandal, "Computational intelligence based triple layer perceptron model coordinated PSO guided metamorphosed based application in cryptographic technique for secured communication (TLPPSO'13)," *Procedia Technology*, vol. 10, no. 1, pp. 433-442, 2013.

[12] W. Sirichotedumrong, T. Chuman, S. Imaizumi, *et al.*, "Grayscale-based block scrambling image encryption for social networking services," in *IEEE International Conference on Multimedia and Expo (ICME'18)*, 2018. arXiv:1806.03787.

[13] L. Teng, H. Li, "A high-efficiency discrete logarithm-based multi-proxy blind signature scheme," *International Journal of Network Security*, vol. 20, no. 6, pp. 1200-1205, Nov. 1, 2018.

[14] X. Y. Wang, L. Yang, R. Liu, *et al.*, "A chaotic image encryption algorithm based on perceptron model," *Nonlinear Dynamics*, vol. 62, no. 3, pp. 615-621, 2010.
[15] G. Ye, K. W. Wong, "An image encryption scheme based on time-delay and hyperchaotic system," *Nonlinear Dynamics*, vol. 71, no. 1-2, pp. 259-267, 2012.
[16] S. Yin, H. Li, and L. Teng, "A new chi-square distribution de-noising method for image encryption," *International Journal of Network Security*, vol. 21, no. 5, pp. 804-811, 2019.
[17] S. Yin, J. Liu and L. Teng, "Improved elliptic curve cryptography with homomorphic encryption for medical image encryption," *International Journal of Network Security*, vol. 22, no. 3, pp. 421-426, 2020.
[18] R. Yoshinaka, M. Kawashima, Y. Takamura, *et al.*, "Adaptive control of robot systems with simple rules using chaotic dynamics in quasi-layered recurrent neural networks," *Studies in Computational Intelligence*, vol. 399, pp. 287-305, 2012.
[19] Y. Zhang, "Test and verification of AES used for image encryption," *3D Research*, vol. 9, no. 1, pp. 3, 2018.
[20] Q. Zhang, Q. Ding, "Digital image encryption based on advanced encryption standard (AES)," in *The Fifth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC'15)*, 2015. DOI: 10.1109/IMCCC.2015.261.
[21] P. Zhang, J. Liu, C. Chen, *et al.*, "The algorithm study for using the back propagation neural network in CT image segmentation," in *International Conference on Innovative Optical Health Science. International Society for Optics and Photonics*, 2017. (`https://doi.org/10.1117/12.2267070`)
[22] C. Zhao, S. Yin, H. Li, and Y. Sun, "Medical image encryption based on stream cipher algorithm and krill group," *International Journal of Network Security*, vol. 22, no. 2, pp. 314-320, 2020.

Biography

Desheng Liu was born in 1979. He received the Ph.D. degree in mechanical design and theory from Northeast Forestry University, China.He is currently a Professor with Jiamusi University, China. His research interests include theInternet of Things, intelligent control, and machine learning (Contact him at email: zdhlds@163.com).

Fuqiang Wang is a master in School of information and electronic technology, Jiamusi University. His research interests include intelligent control, and machine learning (Contact him at email: 1175881343@qq.com).

Hui Wang majored in electrical engineering, Harbin University of technology. Her main research direction is motor analysis and control drive (Contact her at email: wh4922866@163.com).

Campus Wireless Network Coverage and Analysis of Its Security Based on Big Data

Yang Chen, Yingyun Wang, and Fenfei Gu
(Corresponding author: Yang Chen)
Institute of Information Engineering, Anhui Xinhua University, Hefei, Anhui 230088, China
No. 555, Wangjiang West Road, Hefei, Anhui 230088, China
Email: bochen19137542@163.com
(Received Apr. 1, 2019; Revised and Accepted June 24, 2020; First Online June 10, 2021)

Abstract

Under the background of big data, the wired network has not met the campus's needs; therefore, it needs to cover the wireless network. Taking the wireless local area network (WLAN) coverage of Institute of Information Engineering of Anhui Xinhua University as an example, this study firstly analyzed the characteristics of campus WLAN and the security technology of WLAN, improved the wired equivalent privacy (WEP) protocol in IEEE802.11, and strengthened the security of RC4. Then, a specific coverage scheme of campus WLAN was designed. After testing, it was found that the operation efficiency of the improved RC4 algorithm decreased slightly, and the designed WLAN could defend most of the network attacks, had high signal intensity, and operated stably, meeting the needs of teachers and students. Thus, this study contributes to the further research of WLAN and provides some guidance for constructing campus WLAN.

Keywords: Big Data; Security; Wireless Network

1 Introduction

With the development of technology, people's demands on the Internet has improved, and their dependence on wireless local area network also has been strengthened [4, 6]. In the current campus, the wired network is mainly used, which provides great convenience for teachers and students. However, the wired network has some limitations. For example, it can not meet the needs of teachers and students to use the network at any time in the classroom, library, outdoor, etc. Also, with the development of science and technology, equipment, such as tablet computers, mobile phones, and laptops, has become more and more popular.

In order to satisfy the demands of teachers and students, covering WLAN on campus has become an important part of campus construction. With the popularity of WLAN on campus [10], how to build campus WLAN and realize the safe transmission of information has been widely studied. Zhang *et al.* [21] designed a campus network management method based on the runtime model to manage the network equipment uniformly. They found through the experiment that the designed method could save energy by 16.7% and manage the network more efficiently and orderly compared with the traditional method.

Nonum *et al.* [16] studied WLAN in Nigerian tertiary institutions, proposed an autonomous web service architecture, which could not only manage the performance of service users but also manage the interconnection of WiMax-WiFi infrastructure and service coverage network, to improve the network flexibility and overall performance.

Zhang *et al.* [20] designed an encryption-based method to protect user privacy and carried out experiments on the method in the intelligent campus. It was found that the method provided a more powerful security guarantee and weighed the storage, bandwidth, and computing costs, which had high practicability. Ooko *et al.* [18] pointed out that there have been increasing hacking cases because of the uncontrolled medium of WLAN. They investigated the security of WLAN in Kenyan University and found through literature analysis, interviews, and experiments that campus WLAN was not secure.

2 Campus Wireless Network Under Big Data

In the era of big data, the number and types of information have become more and more, which brings great challenges to the security of the network [11]. A school is a place with a large number of people and frequent exchanges. In the campus wireless network, teachers and students are producing data almost every moment. These data packets contain a variety of content, including text, video, audio, etc., and the data contain very important information, such as the identity information, major, and

performance of students, the teaching courseware and research paper of teachers, etc. Mass data have higher requirements on the network; therefore, a network environment with higher processing speed and safety is needed. In campus WLAN, the existing problems mainly include:

Equipment Safety. In order to ensure the physical security of the network, it is necessary to ensure the operation security of wireless devices and detect these devices to avoid equipment failure and protect the wireless network;

Information Security. Under the background of big data, information spreads very fast, and it is easy to leak and lose information. In the process of information transmission, there may be various loopholes, which will cause damages to network security under the attack of viruses or hackers. Also, imperfect and incomplete network management will also cause security problems.

Compared with the wired network, WLAN is more vulnerable and prone to problems because of its openness [7]. Also, wireless devices have some limitations in storage, power supply, etc. Many security technologies that can be applied in the wired network can not be applied in the security protection of WLAN. At present, the main security problems faced by WLAN are as follows.

The wired network has fixed boundaries; therefore, attackers need to go through defense lines, such as firewalls, gateways, etc., to enter the network. WLAN has no clear defense boundary; therefore, attackers can attack the network from any node. Wired network terminals can not move on a large scale, so that it is easy to manage, while WLAN terminals are mobile and vulnerable to eavesdropping and hijacking because of insufficient protection [13]. Also, the topology of WLAN is dynamic and changeable; therefore, it is difficult to carry out centralized management, but many security algorithms require the participation of all nodes, which is difficult to achieve in WLAN [14]. Finally, with the movement of users, the channel of WLAN is affected by interference and fading, leading to a large fluctuation in signal quality, i.e., the robustness problem.

3 Overview of Wireless Network

3.1 Composition of WLAN

WLAN refers to the local area network established using wireless communication [9, 12]. It has fast transmission speed and has been widely used in scenes such as personal, enterprise, school [19]. It takes up a small space and covers a wide range. It can also be used in areas such as forests and deserts with constant communication. The main components of WLAN include:

1) Station (STA, pp. a data exchange equipment, such as desktop, notebook, mobile phone, etc.;
2) Wireless medium (WM, pp. the medium that can transmit radio frequency (RF) signal or infrared signal;
3) Access point (AP, pp. realize the exchange of wireless data;
4) Distributed system: realize the connection between different services.

3.2 WLAN Security Technology

3.2.1 IEEE 802.11 Standard

In order to realize the communication and data sharing of WLAN, it should follow some protocols. WLAN protocols have good security and stability. At present, the most commonly used is the IEEE 802.11 standard [1, 15].

IEEE802.11 is an open-band network with a data transmission rate of 1 Mbit/s-2 Mbit/s and a working frequency band of 2.4 GHz. In addition to IEEE802.11, there are 802.11a, 802.11b, 802.11g, and 802.11n. The latter four networks are the upgrading and improvement of 802.11. Its security technologies include:

1) Service set identification (SSID, pp. SSID is a string. As long as the SSID is remembered, it can access WLAN to avoid unauthorized users accessing the network.
2) WEP: WEP is used for WLAN encryption and authentication [2]. The encryption algorithms used are RC4 (Rivest cipher) and CRC-32. The former is used for ensuring data security, and the latter is used for ensuring data integrity.
3) Physical address filtering (MAC, pp. only the site registered with MAC address can be connected. Each network card corresponds to a unique MAC address, which can prevent some low-level intrusion.
4) Identity authentication: there are two methods: open system authentication (OSA) and shared key authentication (SKA). The former is the default method, and the latter is optional. If the request fails to pass the authentication, it will be rejected.

3.2.2 WEP Protocol and Improvement

In WEP, the data frame consists of three parts, 32-bit initialization vector (IV), transmitted data (≥ 1 bit), and 32-bit integrity check value (ICV). IV is transmitted in plaintext, and the last two parts are transmitted in the ciphertext. WEP uses the RC4 algorithm for encryption [17]. It generates a key sequence based on a large array, called S-box, and its value range is 0-255. The encryption process mainly consists of two parts [3]:

1) Key scheduling algorithm (KSA, pp. a secrete key (Key) is randomly selected. S-box is initialized. It is assumed that there are parameters i_t and j_t pointing

to the S-box. Let i_t traverse every position of the S-box to make j_t generate a new value. Then, the bytes corresponding to j_t and i_t in the S-box are exchanged. After N times of traversal, the initial state S_0 of RC4 is obtained.

2) Pseudo-random key sequence generation algorithm (PRGA, pp. according to S_0, i_t and j_t are initialized. Then j in the algorithm is updated. The bytes corresponding to i_t and j_t are exchanged. The position of the bytes in the S-box is conversed through the pseudo-random number generator (PRNG) [5]. After every time of conversion, the 8-bit key stream is output and processed by xor encryption with the plaintext and by xor decryption with the ciphertext.

The key stream in RC4 is generated by PRNG. By analyzing the first byte of the key stream, the first byte of the original key can be found out, and then the other bytes are gradually deduced [8]. Therefore, in order to increase the security of the algorithm, the process of PRNG generating the initial random number is improved. 0-255 is converted to hexadecimal numbers and arranged in a table of 16×16 in the order from small to large, as shown in Table 1.

Table 1: Initialized random number table

	0	1	2	⋮	E	F
0	00	01	02	⋮	0E	0F
1	10	11	12	⋮	1E	1F
2	20	21	22	⋮	2E	2F
⋮	⋮	⋮	⋮	⋮	⋮	⋮
E	E0	E1	E2	⋮	EE	EF
F	F0	F1	F2	⋮	FE	FF

The improved RC4 method improves security by exchanging bytes. It is assumed that the byte of the position where $S\lfloor i_t \rfloor$ locates is $X_{mt}Y_{mt}$ and the byte of the position where $S\lfloor j_t \rfloor$ locates is $X_{nt}Y_{nt}$. The row is moved to the right gradually, and the distance is $|X_{nt} - X_{mt}|$. When $S\lfloor i_t \rfloor$ and j_t are in the same column, the column is moved downward to make $S\lfloor i_t \rfloor$ reach the position of j_t, and the distance is $|Y_{bt} - Y_{mt}|$, where j_t locates is also moved to make $S\lfloor j_t \rfloor$ reach the position of $S\lfloor i_t \rfloor$ according to the same method; the moving distance of the row and column is $(16 - |X_{nt} - X_{mt}|)$ and $(16 - |Y_{nt} - Y_{mt}|)$ respectively. The other steps are the same as the original RC4 algorithm.

4 Coverage Design of Campus WLAN

By February 2020, there were more than 3400 students and 120 teachers in the Institute of Information Engineering of Anhui Xinhua University, including 24 professional laboratories for wireless sensor network and software engineering, six campus practice bases, 11 computer basic experimental training rooms, six embedded experimental rooms, etc. The original wireless network of the school covered a few apartments and teaching buildings, which could not meet the needs of teachers and students. The requirements for WLAN coverage in different places of the college are shown in Table 2.

In the college, the architecture of thin AP was adopted to manage user data uniformly, and the WLAN supporting 802.11n was constructed. RG-WS5708 product was used as the wireless controller; the controller adopted the MIP64 multi-core processor architecture, which could break through the three-tier network to maintain communication with AP and support 768 wireless access points at most. The wireless AP adopted RG-AP220-E, which could provide sixfold bandwidth. The network management system adopted RG-SNV, which could maintain and manage the network remotely and make a timely response if there was an abnormality in the network. Based on the investigation of the actual situation of the college, it was estimated that 123 AP was needed to cover the whole college.

5 Wireless Network Security Analysis

Firstly, the performance of the improved RC4 algorithm was analyzed. The times of operation was compared between the original RC4 algorithm and the improved RC4 algorithm. The results are shown in Tables 3 and 4.

It was seen from Table 3 that the improved RC4 algorithm needed one more row shift and column shift every time when encrypting one byte compared to the original RC4 algorithm. Table 4 shows the operation times of the two algorithms under different byte numbers. It was found that the operation times of the improved RC4 algorithm were slightly more than that of the original RC4 algorithm, 5000, 10000, 15000 and 20000 times respectively, i.e., when improving the security of the algorithm, the operation efficiency of the algorithm decreased slightly.

The method of active analysis was used to test the security of WLAN. A pretended attacker interacted with STA and AP and then attacked WLAN. Whether WLAN could defend against this attack was determined. Test cases were written using Tcl. The test cases were expanded into commands using the C/C++ interface to simulate the attack behavior. Whether it was successful or not, AP will return the results to the host side. The test results are shown in Table 5.

Table 2: Coverage requirements of campus WLAN

Place	Coverage Requirements	Frequency Band Planning
Dense office area	The number of online users shall not be less than 100% of the seats, and the rate per user shall not be less than 2 Mbps	Dual-frequency
Classroom	The number of online users shall not be less than 30% of the seats, and the rate per user shall not be less than 1 Mbps	Single-frequency 2.4 GHz
Library	The number of online users shall not be less than 60% of the seats, and the rate per user shall not be less than 2 Mbps	Dual-frequency
Laboratory	The number of online users shall not be less than 100% of the seats, and the rate per user shall not be less than 1 Mbps	Single-frequency 2.4 GHz
Restaurant	The number of online users shall not be less than 15% of the seats, and the rate per user shall not be less than 1 Mbps	Single-frequency 2.4 GHz
Student apartment	The number of online users shall not be less than 100% of the seats, and the rate per user shall not be less than 2 Mbps	Single-frequency 2.4 GHz
Outdoor playground	The number of online users shall not be less than 80, and the rate per user shall not be less than 1 Mbps	Dual-frequency

Table 3: Comparison of operation times (1)

	Original RC4 Algorithm	Improved RC4 Algorithm
Number of modular addition operations	$5n$	$5n$
Byte conversion times	$2n$	n
Number of row shift operations	-	n
Number of column shift operations	-	n

Table 4: Comparison of operation times (2)

Number of Bytes	Original RC4 Algorithm	Improved RC4 Algorithm
5000	35000	40000
10000	70000	80000
15000	105000	120000
20000	140000	160000

Table 5: Attack test results

Attack Modes	Attack Results
WEP Share Key attack	The attack failed
WEP Weak Key attack	The attack failed
Association Request Frame Flood attack	The attack failed
Virtual Carrier Sense attack	The attack failed
NAV DOS attack	The attack was successful
Beacon Flood attack	The attack failed
EAP Failure attack	The attack failed
Probe Request Frame Flood attack	The attack was successful
Spoof of Sleep State Indication Frame attack	The attack failed
Spoof of No Data TIM Frame attack	The attack failed

It was seen from Table 5 that only the NAV DOS attack and Probe Request Frame Flood attack were successful, and the other eight attacks failed, which showed that the WLAN established in this study had a good performance in security and could resist most of the attacks. Also, for the successful cases of attacks, managers should pay attention to them and further strengthen the security of WLAN by combining with methods such as intrusion detection.

The other performances of the designed WLAN were tested. The testing content is as follows.

1) The signal strength in the classroom installed with AP was analyzed. A laptop was put in the classroom and installed with the wirelessmon software. The laptop was moved freely in the classroom, and the signal data of different positions were recorded. After testing, it was found that the signal strength in the classroom was - 32 dB ¿ 50 dB, which showed good quality.

2) The download rate of AP was tested. In the classroom, five laptops were connected with AP, and large files were downloaded through Thunder 7. After 20 minutes, the download stopped, and the total download amount was calculated. The test showed that the download rate of AP was 800 kb/s, which could meet the needs of teachers and students.

3) The response of AP in case of failure was tested. In the classroom, large files were downloaded via a laptop connected to AC. Whether the download of the laptop interrupted was tested after two running AC was turned off. After testing, it was found that the download did not interrupt, indicating that the AP could automatically carry out local forwarding and had a stable data forwarding function.

4) The warning function of WLAN was tested. A wireless AP was added as an illegal AP, and its SSID was set as wlanx. Then, a laptop was connected to observe whether it could connect successfully. After testing, the terminal connecting to the AP was forcibly interrupted, indicating that WLAN could prevent the access of illegal AP.

6 Conclusion

This paper analyzed the security problems of WLAN. An improved RC4 algorithm was proposed to enhance the security of WLAN. The campus WLAN coverage scheme was designed, and the WLAN was tested. It was found that the WLAN had good signal strength, a high download rate, a stable data transmission function, a stable alarm function, and a strong defense against network attacks. This work makes some contributions to the construction of campus WLAN.

References

[1] A. S. M. Anuar, W. N. W. Muhamad, D. M. Ali, S. Seroja, N. A. Wahab, "A review on link adaptation techniques for energy efficiency and QoS in IEEE802.11 WLAN," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 1, pp. 331, 2020.

[2] A. Aziz, M. R. Abd Razak, N. E. A. Ghani, "The performance of different IEEE802.11 security protocol standard on 2.4ghz and 5GHz WLAN networks," in *International Conference on Engineering Technology and Technopreneurship (ICE2T'17)*, Kuala Lumpur, pp. 1-7, 2017.

[3] S. Chugh, "Kamal. Securing data transmission over wireless LAN (802.11) by redesigning RC4 Algorithm," in *International Conference on Green Computing & Internet of Things*, pp. 1436-1441, 2015.

[4] Z. Guo, "Cryptanalysis of a certificateless conditional privacy-preserving authentication scheme for wireless body area networks," *International Journal of Electronics and Information Engineering*, vol. 11, no. 1, pp. 1-8, 2019.

[5] D. Han, L. Min, G. Chen, "A stream encryption scheme with both key and plaintext avalanche ef-

fects for designing chaos-based pseudorandom number generator with application to image encryption," *International Journal of Bifurcation & Chaos*, vol. 26, no. 05, pp. 1650091, 2016.

[6] D. P. Huangfu, X. P. Tian, X. J. Wang, P. Chen, "Research and mass deployment of non-cognitive authentication strategy based on campus wireless network," *ITM Web of Conferences*, vol. 17, pp. 01013, 2018.

[7] M. S. Hwang, C. C. Lee, S. K. Chong, J. W. Lo, "A key management for wireless communications", *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 8, pp. 2045–2056, 2008.

[8] R. Ito, A. Miyaji, "Refined construction of RC4 key setting in WPA," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. 100, no. 1, pp. 138-148, 2017.

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

[10] Q. Liao, X. R. Luo, A. Gurung, W. Shi, "A holistic understanding of non-users' adoption of university campus wireless network: An empirical investigation," *Computers in Human Behavior*, vol. 49, pp. 220-229, 2015.

[11] L. Liu, Z. Cao, C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, pp. 29–35, 2018.

[12] J. W. Lo, C. C. Lee, M. S. Hwang, Y. P. Chu, "A secure and efficient ECC-based AKA protocol for wireless mobile communications", *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 11, pp. 5249-5258, 2010.

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

[14] J. W. Lo, S. C. Lin, M. S. Hwang, "A parallel password-authenticated key exchange protocol for wireless environments", *Information Technology and Control*, vol. 39, no. 2, pp. 146-151, 2010.

[15] W. N. W. Muhamad, J. Y. Khan and J. Brown, "Energy efficient contention window adaptation algorithm for IEEE 802.11 WLAN," in *22nd International Conference on Telecommunications (ICT'15)*, pp. 54-59, 2015.

[16] E. O. Nonum, P. O. Otasowie, K. C. Okafor, "Campus wireless network classification for enterprise adoption: perspectives and dimensions for large scale computing," *International Journal of Computer Applications*, vol. 142, no. 12, pp. 19-31, 2016.

[17] T. Ohigashi, T. Isobe, Y. Watanabe, M. Morii, "Full plaintext recovery attacks on RC4 using multiple biases," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. 98, no. 1, pp. 81-91, 2015.

[18] O. S. Ooko, M. Shadrack, E. Ataro, "Security of wireless campus networks in selected public and private universities in Kenya," *International Journal of Engineering and Management Sciences*, vol. 2, no. 1, pp. 1-10, 2017.

[19] A. Pandey, P. K. Pant, R. C. Tripathi, "A system and method for authentication in wireless local area networks (WLANs)," *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, vol. 86, no. 2, pp. 149-156, 2016.

[20] L. Q. Zhang, O. Oksuz, L. Nazaryan, C. Q. Yue, B. Wang, A. Kiayias, A. Bamis, "Encrypting wireless network traces to protect user privacy: A case study for smart campus," in *IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'16)*, New York, NY, pp. 1-8, 2016.

[21] P. Zhang, J. Wang, "Management of intelligent campus wireless sensor networks based on runtime model," *Journal of Computer & Communications*, vol. 03, no. 7, pp. 22-31, 2015.

Biography

Yang Chen, born in June 1980, has received the master's degree from Hefei University of Technology. She is an associate professor. Her research directions are wireless sensor network, network security, and artificial intelligence.

Yingyun Wang, born in November 1982, has received the master's degree from Anhui University. She is an associate professor. She is interested in network security and artificial intelligence.

Fenfei Gu, born in September 1982, has received the master's degree from Hefei University of Technology. He is associate professor. He is interested in deep learning and privacy protection.

Research on Network Intrusion Recognition Based on an Intelligent Algorithm

Shuo Wang

(Corresponding author: Shuo Wang)

School of Energy and Intelligence Engineering, Henan University of Animal Husbandry and Economy
No. 16, Beilin Road, Jinshui District, Zhegnzhou 450044, China
Email: swgxx008@163.com

(Received Jan. 21, 2019; Revised and Accepted Sept. 18, 2020; First Online June 24, 2021)

Abstract

Timely identification of network intrusion plays an important role in the realization of network security. Based on the intelligent algorithm, this study analyzed network intrusion identification. First, based on the back-propagation neural network (BPNN) model, ACO-BPNN and PSO-BPNN models were obtained by optimizing the parameters by ant colony optimization (ACO) and particle swarm optimization (PSO) algorithms. Then, the model was tested on the KDD CUP99 data set. The results showed that the accuracy and precision of the model optimized by the intelligent algorithm greatly improved; the PSO-BPNN model had the best performance, with an accuracy of 95.39%, a precision of 87.13%, a recall rate of 91.4%, and an F1 score of 60.59%, and spent a short time in recognition, 0.89 s, showing a good performance in intrusion recognition. Thus, the results showed that the intelligent algorithm plays an important role in network intrusion identification, which is conducive to the further improvement of the efficiency of intrusion identification and the realization of network security.

Keywords: Network Security; Intrusion Recognition; Intelligent Algorithm; Neural Network

1 Introduction

With the development of the network, it has been applied more and more extensively and become more and more popular, the number of network users also increases [18]. As information becomes larger and more and more complex [8], the network creates excellent conditions for the circulation and sharing of information, which brings great convenience for people's life and work [2]. However, due to the inherent openness of the network, the network security problem is becoming more and more serious [27], such as information alteration, impersonation, and hacker intrusion [14].

With the deepening of government informatization, information security not only affects individuals but also involves society, government, and even the country. To achieve network security, many technologies have been applied, such as the firewall system [17], the identity authentication system [9, 16, 24], *etc.*, but they are usually based on defense and can not identify and detect network intrusion timely. Therefore, the method of network intrusion identification has been widely concerned by researchers [30]. Subba *et al.* [19] designed an intrusion detection method based on game theory, which prevented malicious behavior between nodes using two different reputation updating and expulsion mechanisms and based on Shapley value mechanism and Vickery-Clark-Grooves (VCG) mechanism. Through simulation, they found that the method had high accuracy and detection rate in the attack range.

Ali *et al.* [1] designed a fast learning network (FLN) model based on particle swarm optimization (PSO), namely the PSO-FLN model, and verified it on KDD99 data set. They found that the PSO-FLN model had significant advantages in accuracy than classifiers such as extreme learning machine (ELM) and FLN. Vahid *et al.* [23] proposed a hybrid algorithm based on K-means and multiple classifiers. Firstly, the data were partitioned by the K-means algorithm and classified by naive Bayes, support vector machine (SVM), and oneR classification algorithm. The results showed that the hybrid method had a detection rate of 99.5%.

Teng *et al.* [22] proposed a method combining decision tree (DT) [6] and SVM to realize the network adaptive writing intrusion recognition and verified the feasibility and effectiveness of the method on the KDD CUP99 data set. This study mainly analyzed the back-propagation neural network (BPNN) model and the optimization effect of ant colony optimization (ACO) and PSO algorithms on the BPNN model and compared different models to understand their performance in intrusion identification. This work makes some contributions to the further development of network intrusion identification.

2 Application of BPNN in Intrusion Recognition

Intelligent algorithm refers to the algorithm created by imitating natural structures under the inspiration of natural laws, including the artificial neural network (ANN) algorithm [7,13], genetic algorithm (GA) [20], swarm intelligence algorithm [3], *etc.* Swarm intelligence algorithms refer to a series of algorithms generated by simulating social insect behaviors, including ant colony optimization algorithm (ACO) [12], particle swarm optimization (PSO) [11], *etc.* The swarm intelligence algorithm generally has good robustness and is easy to realize, which has become an important direction of computer research. This paper mainly analyzed the application of BPNN, ACO, and PSO algorithms in network intrusion recognition.

ANN simulates the structure of the human brain nervous system. It has good nonlinear approximation ability, fast calculation, and excellent learning ability for uncertain systems. It has been widely used in pattern recognition, artificial intelligence, *etc.* After training, a neural network can predict the information when the input value is given, which is very suitable for intrusion recognition. In this study, BPNN, a kind of ANN, is used for intrusion recognition [5]. It is assumed that the input data is $X = (x_1, x_2, \cdots, x_n)$ and the output sample is $Y = (y_1, y_2, \cdots, y_n)$. For a three-layer BPNN, its learning sample is (X_k, C_k), the output of the input layer is Y_i, the output of the hidden layer is Y_j, and the output of the output layer is Y_m. The weight is represented by w, and the threshold is represented by b. The algorithm trains the network by correcting errors. The calculation methods of outputs and errors in different layers are as follows.

The output of the hidden layer is:

$$Y_j = f(\sum_{i=1}^{n} w_{ij} Y_i - b_j).$$

The output of the output layer is:

$$Y_k = f(\sum_{j=1}^{n} w_{jk} Y_j - b_k).$$

The error of the output layer:

$$\varepsilon_k = -(C_k - Y_k) Y_k (1 - Y_k).$$

The error of the hidden layer:

$$\varepsilon_j = Y_j (1 - Y_j) \sum_{k=1}^{m} \varepsilon_k w_{jk}.$$

The correction formulas of weight and threshold are:

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \eta \varepsilon_j Y_i \\ b_j(t+1) &= b_j(t+1) + \eta \varepsilon_j. \end{aligned}$$

Through these two formulas, the error is continuously corrected. When the error reaches the set accuracy, the network training finishes.

However, BPNN has some defects in the application process. It is easy to fall into local minimum and converges slowly. In this study, BPNN was improved by optimizing the weight and threshold with the swarm intelligence algorithm.

3 BPNN Model Combined With Swarm Intelligence Algorithm

3.1 ACO-BPNN Intrusion Recognition Model

The ACO algorithm is a simulation of ant foraging behavior. In the process of foraging, ants will leave pheromones on the way. The shorter the path is, the higher the pheromone concentration is; moreover, the path with higher pheromone is more likely to be selected by ants. Therefore, the shortest path can be selected. The optimal weight and threshold in the BPNN model can be found out using the ACO algorithm. It is assumed that there is a problem to be solved,

$$b = \min f(a),$$

where $f(a)$ stands for the fitness value. The initial pheromone of ant t can be expressed as:

$$\Delta\delta(t) = \exp(-f(a_t)), t = 1, 2, \cdots, M.$$

Its initial position can be written as:

$$A_t = (a_{t_1}, a_{t_2}, \cdots, a_{t_d}).$$

If $f(A_t) \geq 0$, then $\Delta\delta(t) \in (0, 1]$. $f(a)$ can be written as

$$f'(A_t) = \begin{cases} \frac{f(A_t)}{avg} & avg > 0 \\ f(A_t) & \text{otherwise} \end{cases}$$

after transformation.

In the process of optimization, the number of ants selected is e, $e = \lfloor r, M \rfloor$, where r refers to the selection ratio. It is assumed that the objective of the optimization is A_{ocj} and the optimal solution obtained from the last search is A_{best}; then,

$$A_{ocj} = \begin{cases} A_j & \delta(A_t) < \max \delta(A_j) \\ A_{best} & \text{otherwise} \end{cases}$$

The specific search method can be written as:

$$A_{best} = \begin{cases} A'_t & f(A'_t) < f(A_{best}) \\ A_{best} & \text{otherwise} \end{cases}$$

The updating formula of pheromone can be written as:

$$\delta(t) = (1 - lambda)\delta(t) + \Delta\delta(t),$$

where λ is the volatility coefficient.

The flow of the ACO-BPNN intrusion recognition model is as follows. The parameters of the ACO algorithm are initialized, and then the weight and threshold of the BPNN model are optimized. After reaching the maximum number of iterations, the optimal parameters are output and input into the BPNN model. Then, the BPNN model is trained using the intrusion training samples. After training, the model is used for recognizing network intrusion behaviors.

3.2 PSO-BPNN Intrusion Recognition Model

The PSO algorithm is a simulation of birds foraging behavior. It generates a group of examples randomly and flies randomly in the search space to find the optimal particle position. The algorithm is simple and easy to implement, which has good applications in parameter optimization, system control, *etc.* [21].

The PSO algorithm mainly updates the position by two values: individual extremum P_i and global extremum P_g. It is assumed that there are n particles in a d-dimensional search space, the position and speed of the i-th particle at time t are $X_i = (x_{i_1}, x_{i_2}, \cdots, x_{i_d})^t$ and $V_t = (v_{i_1}, v_{i_2}, \cdots, v_{i_d})^t$ respectively. The individual extremum is $P_i = (p_{i_1}, p_{i_2}, \cdots, p_{i_d})^t$, and the global extremum is $P_g = (p_{g_1}, p_{g_2}, \cdots, p_{g_d})^t$. The updating formulas of speed and position are:

$$\begin{aligned} v_{i_d}^{t+1} &= w v_{i_d}^t + c_1 r_1 (p_{i_d}^t - x_{i_d}^t) + c_2 r_2 (p_{g_d}^t - x_{g_d}^t) \\ x_{i_d}^{t+1} &= x_{i_d}^t + v_{i_d}^t, \end{aligned}$$

where w is the inertia weight, c_1 and c_2 are learning factors, and (r_1, r_2) are random numbers in $\lfloor 0, 1 \rfloor$.

The flow of the PSO-BPNN intrusion recognition model is as follows. The parameters of the PSO algorithm are initialized. The fitness value of particles is calculated. The particles are compared with P_i and P_g and updated until the optimal particles are obtained. The optimal particles were input into the BPNN model as the optimal weight and threshold of the BPNN model. Then, the trained BPNN model is used for recognizing intrusion behaviors.

4 Experimental Analysis

4.1 Experimental Data Set

The model designed in this study was tested by the KDD CUP99 data set [4]. In the KDD CUP99 data set, each intrusion was described by 41 kinds of features. The data set includes four types of intrusion:

1) DOS: making the normal service of legitimate users cannot respond;

2) Probe: taking advantages of system vulnerabilities to scan ports and guess passwords;

3) U2R: sending data packets to test the access right;

4) R2l: illegal access to a remote machine.

The KDD CUP99 data set provided a 10% training set and a test set and a testing set, as shown in Table 1.

Table 1: Experimental data set

Intrusion Type	Training Set	Testing Set
Normal	97278	60593
DOS	391458	229853
Probe	4107	4166
U2R	52	228
R2L	1126	16189

4.2 Evaluation Index

After model recognition, the actual situation and recognition situation of samples are represented by the confusion matrix, as shown in Table 2.

Table 2: Confusion matrix

Actual Situation	Recognition Results	
	Positive Sample	Negative Sample
Positive Sample	TP	FN
Negative Sample	FP	TN

The evaluation indexes of the model are as follows. Accuracy:

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

Precision:

$$Precision = \frac{TP}{TP + FP}$$

Recall rate:

$$Recall = \frac{TP}{TP + FN}$$

F1 score:

$$F1_{score} = \frac{2TP}{2TP + FP + FN}$$

F1 score was the harmonic average of precision and recall rate; the higher the score was, the better the performance of the model was.

4.3 Recognition Results

The performance of BPNN, ACO-BPNN, and PSO-BPNN models in recognizing intrusion categories was

compared. The recognition results of different models are shown in Tables 3, 4, and 5.

It was seen from Tables 3, 4, and 5 that the number of TP and TN gradually increased, while the number of FP and FN gradually decreased, indicating that the performance of the latter model was better than that of the former model. TP refers to the situation that positive samples were recognized as positive samples, while FN refers to the situation that negative samples were recognized as negative samples. The number of TP and FN recognized by the PSO-BPNN model was larger than that of BPNN and ACO-BPNN models, while the number of FP and FN was smaller. To further clarify the performance of the models, the evaluation indexes were calculated and compared, and the results are shown in Figure 1.

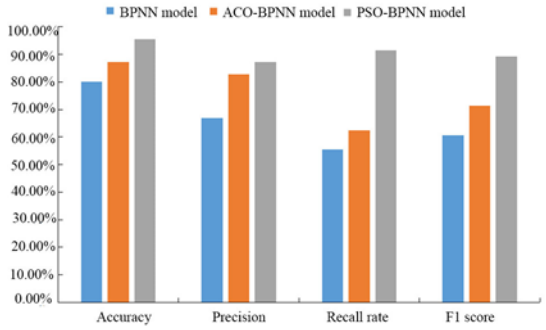


Figure 1: Comparison of performance between models

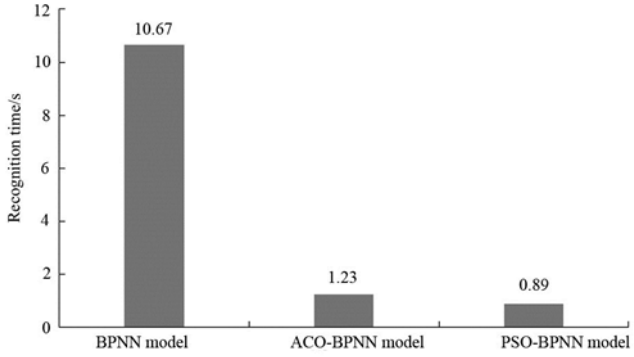


Figure 2: Comparison of recognition time between models

As shown in Figure 1, the accuracy of the three models was 80.08%, 87.15%, and 95.39%, respectively, i.e., the accuracy of the PSO-BPNN model was the highest, which was 15.31% higher than BPNN and 8.24% higher than the ACO-BPNN model. The precision of the three models was 66.81%, 82.84%, and 87.13%, respectively, i.e., the precision of the PSO-BPNN model was the highest, which was 20.32% higher than the BPNN model and 4.29% higher than the ACO-BPNN model. The recall rate of the three models was 55.43%, 62.50%, and 91.40%, respectively, i.e., the recall rate of the PSO-BPNN model was the highest, which was 35.97% higher than the BPNN model and 28.90% higher than the ACO-BPNN model. The F1 score of the three models was 60.59%, 71.25%, and 89.21%, respectively, i.e., the F1 score of the PSO-BPNN model was the highest, which was 10.66% higher than the BPNN model and 17.96% higher than the ACO-BPNN model. In conclusion, the performance of the PSO-BPNN model was the best.

Table 3: Recognition results of the BPNN model

	Normal	DOS	Probe	U2R	R2L	Total
TP	15087	27841	146	30	4521	47625
TN	14047	175246	2465	132	9568	201458
FP	11250	10658	692	30	1024	23654
FN	20209	16108	863	36	1076	38292

Table 4: Recognition results of the ACO-BPNN model

	Normal	DOS	Probe	U2R	R2L	Total
TP	15502	28965	154	38	4862	49521
TN	23520	185472	2617	146	9786	221541
FP	1125	7548	569	27	987	10256
FN	20446	7868	826	17	554	29711

Table 5: Recognition results of the PSO-BPNN model

	Normal	DOS	Probe	U2R	R2L	Total
TP	23695	30216	168	46	5120	59245
TN	25058	198524	3124	168	10584	237458
FP	7821	495	216	7	215	8754
FN	4019	618	658	7	270	5572

The recognition time of different models was compared, as shown in Figure 2.

As shown in Figure 2, the recognition time of the BPNN model was 10.67 s but significantly decreased after optimization by the intelligent algorithm; the recognition time of the ACO-BPNN model was 1.23 s, which was 9.44 s less than that of the BPNN model; the recognition time of the PSO-BPNN model was 0.89 s, which was 9.78 s less than that of the BPNN model. It was found that the PSO-BPNN model not only had higher accuracy and recognition rate but also has shorter recognition time, i.e., the PSO-BPNN model had a high recognition efficiency. Therefore, among the three models, the PSO-BPNN model had the best performance in recognizing network intrusion.

5 Discussion

Intrusion refers to all internal and external behaviors that attempt to destroy the security, integrity, and confidentiality of the network, including trying to break in, impersonating legitimate users, malicious use, *etc.* Intru-

sion recognition uses various means to collect user data inside and outside the network for analysis and recognize abnormal behaviors [10]. The existing methods are

1) Colored Petri net [29]: the intrusion behavior is represented by the colored Petri net, which has a simple concept but has a high calculation cost;
2) Expert system [26]: the intrusion behavior is analyzed by experts according to experience;
3) Neural network [28]: the neural network model is established and trained by historical data;
4) Probability and statistics [15]: the normal user behavior model is established and compared to find out abnormal behaviors, which is convenient for maintenance, but can not achieve real-time recognition;
5) Immunity [25]: detect "own" and "non-own" antigens using natural immune system technology. In this study, based on the neural network, intrusion recognition was analyzed.

According to the experimental results, it was found that the original BPNN model had a poor performance in intrusion recognition. After the optimization of the intelligent algorithm, it was seen from Figure 1 that the performance of ACO-BPNN and PSO-BPNN models improved significantly. The indicators of the BPNN model were mostly below 80%; the accuracy of the ACO-BPNN model was 87.15%, but the F1 score of the PSO-BPNN model was only 71.25%, which was only 10.66% higher than that of the BPNN model. The accuracy of the PSO-BPNN model was 95.39%, and the F1 score was 89.21%, which was 17.96% higher than the ACO-BPNN model, i.e., it had a better performance in intrusion detection. The comparison of the recognition time demonstrated that the addition of the intelligent algorithm shortened the time of intrusion recognition. In the intelligent algorithm, the work of each individual is very simple, and the working time is also short. Therefore, while optimizing the performance of the BPNN model in intrusion recognition, it also improved its efficiency.

In this study, although some achievements have been made in the research of network intrusion recognition, there are some shortcomings. In future works, it is necessary to

1) Carry out experiments in the actual running environment of the network;
2) Study the application of more intelligent algorithms;
3) Further improve the effect of intrusion recognition.

6 Conclusion

In this study, the application of the intelligent algorithm in network intrusion recognition was analyzed. The BPNN model was optimized by ACO and PSO algorithms to obtain the ACO-BPNN model and PSO-BPNN model. These models were compared and analyzed. Through experiments, it was found that the BPNN model optimized by the intelligent algorithm had greatly improved performance and correctly recognized more samples, showing higher accuracy and precision. The comparison showed that the PSO-BPNN model had better performance and spent a short time in recognition; therefore, it is more suitable for recognizing network intrusion and can be further promoted and applied in practice.

References

[1] M. H. Ali, B. A. D. Al Mohammed, M. Ismail, M. F. Zolkipli, "A new intrusion detection system based on fast learning network and particle swarm optimization," *IEEE Access*, vol. 99, pp. 1-1, 2018.
[2] M. Alkasassbeh, "An empirical evaluation for the intrusion detection features based on machine learning and feature selection methods," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 22, pp. 5962-5976, 2017.
[3] S. Cheng, Q. Zhang, Q. Qin, "Big data analytics with swarm intelligence," *Industrial Management & Data Systems*, vol. 116, no. 4, pp. 646-666, 2016.
[4] R. Devi, R. K. Jha, A. Gupta, S. Jain, P. Kumar, "Implementation of intrusion detection system using adaptive neuro-fuzzy inference system for 5G wireless communication network," *AEU - International Journal of Electronics and Communications*, vol. 74, pp. 94-106, 2017.
[5] A. Dewanje and K. A. Kumar, "A new malware detection model using emerging machine learning algorithms," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 24–32, 2021.
[6] I. El-Henawy, H. M. El Bakry, H. M. El Hadad, "A new muzzle classification model using decision tree classifier," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 12–24, 2017.
[7] T. T. Gao, H. Li, and S. L. Yin, "Adaptive convolutional neural network-based information fusion for facial expression recognition," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 17-23, 2021.
[8] G. P. Gupta, M. Kulariya, "A framework for fast and efficient cyber security network intrusion detection using apache spark," *Procedia Computer Science*, vol. 93, pp. 824-831, 2016.
[9] M. S. Hwang, J. W. Lo, S. C. Lin, "An efficient user identification scheme based on ID-based cryptosystem", *Computer Standards & Interfaces*, vol. 26, no. 6, pp. 565–569, 2004.
[10] O. Isaiah, A. Olutola, O. Olayemi, "Feature or attribute extraction for intrusion detection system using gain ratio and principal component analysis

(PCA)," *Communications on Applied Electronics*, vol. 4, no. 3, pp. 1-4, 2016.
[11] N. K. Jain, U. Nangia, J. Jain, "A review of particle swarm optimization," *Journal of the Institution of Engineers*, vol. 99, no. 4, pp. 1-5, 2018.
[12] M. Kefayat, A. L. Ara, S. A. N. Niaki, "A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources," *Energy Conversion & Management*, vol. 92, pp. 149-161, 2015.
[13] P. Koprinkovahristova, V. Mladenov, N. K. Kasabov, "Artificial neural networks," *European Urology*, vol. 40, no. 1, pp. 245, 2015.
[14] C. Q. Li, Z. Yan, Y. L. Fu, H. L. Chen, "Data fusion for network intrusion detection," *Security & Communication Networks*, vol. 2018, pp. 1-16, 2018.
[15] P. Li, Z. Wang, H. Xu, F. Zhu, R. Wang, "Intrusion detection methods based on incomplete RFID traces," *Chinese Journal of Electronics*, vol. 26, no. 04, pp. 675-680, 2017.
[16] C. H. Ling, C. C. Lee, C. C. Yang, and M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN", *International Journal of Network Security*, vol. 19, no. 2, pp. 177-181, Mar. 2017.
[17] R. K. Mohammed, Y. Ueno, "An FPGA-based network firewall with expandable rule description," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 3, pp. 1310-1318, 2018.
[18] V. Shah, A. K. Aggarwal, N. Chaubey, "Performance improvement of intrusion detection with fusion of multiple sensors," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 33-39, 2017.
[19] B. Subba, S. Biswas, S. Karmakar, "A game theory based multi layered intrusion detection framework for wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 25, no. 4, pp. 1-23, 2018.
[20] J. D. Szustakowski, Z. Weng, "Protein structure alignment using a genetic algorithm," *Proteins Structure Function and Bioinformatics*, vol. 38, no. 4, pp. 428-440, 2015.
[21] Y. Tang, X. Guan, "Parameter estimation for time-delay chaotic system by particle swarm optimization," *Chaos Solitons & Fractals*, vol. 40, no. 3, pp. 1391-1398, 2017.
[22] S. Teng, N. Wu, H. Zhu, L. Teng, W. Zhang, "SVM-DT-based adaptive and collaborative intrusion detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 001, pp. 108-118, 2018.
[23] S. Vahid, M. Ahmadzadeh, "KCMC: A hybrid learning approach for network intrusion detection using k-means clustering and multiple classifiers," *International Journal of Computer Applications*, vol. 124, no. 9, pp. 18-23, 2015.
[24] H. Yang, W. Cai, Y. Xia, O. Ouyang, X. Xie, "Identity authentication system for mobile terminal equipment based on SDN network," *International Journal of Information and Communication Technology*, vol. 17, no. 3, pp. 257, 2020.
[25] C. Yin, L. Ma, L. Feng, "Towards accurate intrusion detection based on improved clonal selection algorithm," *Multimedia Tools & Applications*, vol. 76, pp. 1-14, 2017.
[26] J. R. Yost, "The march of IDES: Early history of intrusion-detection expert systems," *IEEE Annals of the History of Computing*, vol. 38, no. 4, pp. 42-54, 2016.
[27] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, no. C, pp. 25-37, 2017.
[28] W. Zheng, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 99, pp. 1-1, 2018.
[29] S. J. Zhou, Z. G. Qin, F. Zhang, J. D. Liu, "An attack modeling based on colored petri net," *Journal of Electronic Science and Technology of China*, vol. 2, no. 1, pp. p.47-52, 2004.
[30] R. Zuech, T. M. Khoshgoftaar, R. Wald, "Intrusion detection and big heterogeneous data: A survey," *Journal Of Big Data*, vol. 2, no. 1, pp. 3, 2015.

Biography

Shuo Wang, born in 1974, received the master's degree from Nankai University in 2011. He is an associate professor and working in School of Energy and Intelligence Engineering, Henan University of Animal Husbandry and Economy. He is interested in computer.

Anomaly Detection Based on Discriminative Generative Adversarial Network

Benjamin Appiah[1], Zhiguang Qin[1], Obed Tettey Nartey[2],
Brighter Agemang[2], and Ansuura JohnBosco Aristotle Kanpogninge[2]
(Corresponding author: Benjamin Appiah)

School of Information and Software Engineering, University of Electronic Science and Technology of China[1]
Chengdu, 610054, China
School of Computer Science and Software Engineering, University of Electronic Science and Technology of China[2]
Email: 1746627510@qq.com
(Received Feb. 15, 2020; Revised and Accepted Sept. 8, 2020; First Online June 12, 2021)

Abstract

Generative adversarial network (GAN) has recently achieved remarkable success in anomaly detection. The GAN structure consists of a generator as an attacker and a discriminator as a classifier, all competing against each other. Most of the available classifiers are trained on softmax cross-entropy (SCE) loss function. However, the SCE loss is less discriminative in classifying malicious samples. To improve the discriminative power of these GAN's classifiers, this paper proposes a discriminative GAN, D-GAN, for short. The key idea is to replace the SCE loss with Mahalanobis distance loss to induce strong intra-class compactness to construct high-density regions, which are essential to discriminating against new unseen classes. We evaluate D-GAN on MNIST and KDD99 datasets. The results demonstrate that D-GAN can achieve superior performance compared with several state-of-the-art SCE loss learning-based defending methods.

Keywords: Anomaly Detection; Distance Metric Learning; Generative Adversarial Network

1 Introduction

Generative adversarial network (GAN) has achieved success in modeling complex and high dimensional distributions [9]. Standard Generative Adversarial Networks are composed of a generator (G) and a discriminator (D), which are trained simultaneously by way of adversarial process: The objective of the generator is to capture the data distribution on the other hand the discriminator tries to differentiate between the generated samples and real samples.

To the best of our knowledge, existing GAN-based approaches for anomaly detection [2, 15, 19, 23, 26, 27] adopt softmax cross entropy (SCE) loss (in this work, we term all GAN trained using SCE loss as Softmax GAN) in their discriminator training process. Despite GAN with SCE loss success in anomaly detection, SCE loss as classifier is not robust in classifying malicious samples [3, 10, 12, 22].

During training of the GAN discriminator, in order to effectively classify malicious samples, the deeply learned features are required to discriminate and generalized enough for identifying malicious classes without label predictions. Discriminative power characterizes features in both the compact intra-class variations and separable inter-class differences. However, SCE loss only encourages the separability of features. The resulting features are not sufficient enough for anomaly detection [24, 28].

To resolve the aforementioned issue, [4, 7, 17, 20, 21, 24, 25, 28] focused on means to measure how close the model distribution and real distribution are and try to use discriminative or distance metrics as their objective functions to improve the training of GANs or in other deep learning models that employs the SCE loss function. Discriminative metric learning aims to learn effective metrics to measure the closeness of the input image pairs. For instance, [21] propose a center loss, to efficiently enhance the discriminative power of the deeply learned features in deep learning models. In the course of training their classifiers, they simultaneously update the center and minimize the distances between the deep features and their relative class centers. Nevertheless, these works were not directly applied in GAN for anomaly detection settings.

Inspired by these works, we propose a discriminative GAN, D-GAN for short. We enhance the discriminative power of D-GAN by using the Mahalanobis distance loss as replacement of the SCE loss. Mahalanobis distance loss induce high-density regions in the feature space and learn more structured representations. In the high feature space, the real samples should be close to each other, while further away from the malicious samples, thus, minimizing the inter real samples distance and maximizing real and malicious samples distance. Hence, D-GAN learns to

discriminate and generalized enough for identifying malicious samples. Now, we will describe the layout for the remaining of this paper. In Section 2 , we will briefly review related work. The D-GAN method is described in Section 3. In Section 4, we investigate the performance of D-GAN, and KDD99 [16], and describe the findings. Specifically, the findings demonstrate the superior performance of D-GAN over SCE loss based models. Finally, we conclude the paper in the last Section 5.

2 Related Work

There have been much research in anomaly detection based on GAN which have led to many different proposed techniques [2,15,18,19,23,26,27]. The authors in [23] first trained the GAN on the normal images, and for a test image, the latent space is repetitively searched to find the latent vector that best reconstructs the test image. The authors in [26] leverage the ideas proposed by [5,6] to develop a classifier that is efficient at test time. The authors in [27], propose a bi-directional GANs [5] based anomaly detection. Their model extracts adversarially learned features for the anomaly detection task and uses reconstruction errors based on these adversarially learned features to determine if a data sample is anomalous.

The methods discussed above train GAN with the usual softmax cross entropy loss. Under the usual softmax cross entropy loss, the discriminator was found to be less discriminative, making them not robust enough [3, 4, 7, 10, 12, 17, 20–22, 24, 28]. A more recent method, called Fence GAN [19], address this discriminative shortfall by training the GAN discriminator in the usual adversarial manner with a modified objective. Contrary to these methods, We address this issue by replacing the softmax loss with a distance metric learning loss. Mahalanobis distance loss induce high-density regions in the feature space and learn more structured representations. In the high feature space, the real samples should be close to each other, while further away from the malicious samples, thus, minimizing the inter real samples distance and maximizing real and malicious samples distance.

3 Proposed Method

In this section, a brief introduction of the softmax GAN and present its limitation. Inspired by the analysis, we propose the Mahalanobis distance loss to improve the discriminative power of GAN classifiers.

3.1 Softmax GAN

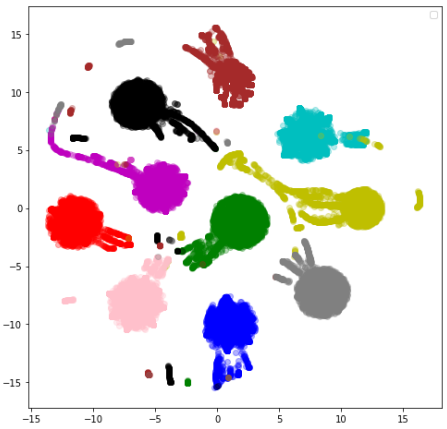
Before we introduce the proposed loss, we first review the Softmax GAN and presents its limitation. Given the real input data B_+ with x samples and the malicious data B_- from generator with $x^{'}$ samples, such that the total number of inputs are $B = B_+ + B_-$. The softmax function; $softmax(h); R^L \longrightarrow R^L$ is represented as $softmax(h_i) = exp(h_i)/\sum_{l=1}^{L} exp(h_i)$, $i \in [L]$, where $L := \{1, ..., L\}$ is the total number of classes and h is termed as logit, i.e, $h = H(z) \in R^L$, $h = Wz + b$, $z = Z(x) \in R^d$. The parameter W and b are the weighted matrix and bias respectively. The loss function of Softmax GAN's discriminator D is defined as:

$$L_D = \sum_{x \in B+} \frac{1}{|B_+|} h(z) + ln \sum_{x \in B} exp^{h(z)} \quad (1)$$

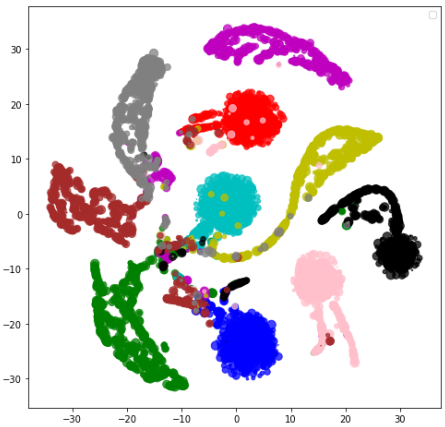
The generator G loss is also expressed as:

$$L_G = \sum_{x \in B+} \frac{1}{|B|} h(z) + \sum_{z^{'} \in B-} \frac{1}{|B|} h(z^{'}) + ln \sum_{x \in B} exp^{h(z)} \quad (2)$$

Despite SCE loss effectiveness and popularity, it has its limitations which is illustrated experimentally in Figure 1. We demonstrate the shortfall of the SCE loss in both the discriminator and the generator in D-GAN network shown in Table 1 on MNIST dataset [16].



(a) training



(b) testing

Figure 1: The t-SNE representation of the deeply learned features in (a) training set (b) testing set, via softmax cross entropy loss on MNIST dataset. The classes in the MNIST dataset are represent with the different colors. Epochs = 100.

The training set and test set consists of 80% and 20% respectively. From Figure 1, it can be seen that, in (a)

the features learned are separable, whiles, in (b) the inter-class distance is not wide enough. During training of the GAN discriminator, in order to effectively classify malicious samples, the deeply learned features are required to discriminate (i.e, wider inter-class distance) and generalize, however, we can see that SCE loss only encourages the separability of features. This limitation leads to poor generalization ability and classification accuracy.

3.2 Mahalanobis Distance

Mahalanobis distance (MD) is an extremely useful metric; Have excellent applications in multivariate anomaly detection, classify highly imbalanced datasets and multi-class classification. Unlike the SCE loss, the Mahalanobis distance learns more structured representations and inducing high-density regions into low-density regions such that these regions are reshaped to maximize the class separability [21].

Given a trainable means of a distribution and sample covariance matrices μ and S respectively. Whereby S can be decomposed as $S = I^T I$, $I \in R^{p\times d}$ and $p \leq d$. The Mahalanobis distribution can be define as:

$$\begin{aligned} MD(x,\mu) &= \sqrt{(I(x-\mu))^T(I(x-\mu))} \\ &= \sqrt{(x-\mu)^T II^{-1}(x-\mu)} \\ &= \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \end{aligned}$$

Where MD is a non-negative and symmetric matrix which can be be shorten as $MD(x,\mu) = \frac{1}{2}\|(x-\mu)\|_2$.

3.2.1 Estimation of the Covariance

The classical Mahalanobis distance discussed above has sub-optimal performance corresponding to the singularity and instability of the sample covariance matrix (S). We address this singularity and instability of the sample covariance matrix using the ideas introduced by [14]. We define a Frobenius norm of a matrix $\|A\| = \sqrt{tr(AA')}$ and estimate the linear combination $\Sigma^* = L_i I + L_2 S$, such that the expected quadratic loss $E[\|\Sigma^* - \Sigma\|^2]$ become minimum. Ledoit and Wolf, [14] proved that the solution for $minL_1L_2E[\|\Sigma^* - \Sigma\|^2]$ satisfies

$$\Sigma^* = \frac{\beta^2}{\delta^2}\mu I + \frac{\alpha^2}{\delta^2}S$$

and

$$E[\|\Sigma^* - \Sigma\|^2] = \frac{\alpha^2\beta^2}{\delta^2}$$

Where $\alpha^2 = \|\Sigma - \mu I\|^2$, $\beta^2 = E[\|S - \mu I\|^2]$, $\delta^2 = E[\|S - \mu I\|^2]$ and $\alpha^2 + \beta^2 = \delta^2$. Ones Σ^* is estimated, the MD can be constructed by replacing S with Σ^*.

3.3 Discriminative GAN

Based on the Mahalanobis distance with stable covariance, we propose a discriminative GAN (D-GAN) for anomaly detection. The D-GAN adopts the Mahalanobis distance to induce high-density regions in the feature space and learn more structured representations. In the high density regions in the feature space, the real samples should be close to each other, while further away from the malicious samples, thus, minimizing the inter real samples distance and maximizing real and malicious samples distance. Hence, D-GAN learns to discriminate and generalized enough for anomaly detection task. The Mahalanobis distance loss function for a single input-label pair $(Z(x), y)$ without softmax is defined as:

$$\ell_{MD}(Z(x), y) = \frac{1}{2}\|(x - \mu_y)\|_2^2$$

The squared-error (i.e, $\ell_{MD}(x,y) = \frac{1}{2}\|(z-\mu_y)\|_2^2$) form is adopted. Using squared-error form can lead to high accuracy with faster convergence rate compared to other forms, since square-error form is not influenced by outliers during the classification of these examples [8].
In D-GAN, the task of the discriminator is to identify whether the testing data conforms the normal data distribution in B_+, the non-conformation parts B_- are classified as attacks, therefore, assign with 0 probability.

$$\begin{aligned} t_D(x) &= \begin{cases} \frac{1}{|B_+|}, & \text{if } x \in B_+ \\ 0, & \text{if } x \in B_- \end{cases} \\ \ell_D(Z(x), y) &= t_D(x)\frac{1}{2}\|(z-\mu_y)\|_2^2 \\ &= \frac{1}{2|B_+|}\|(z-\mu_y)\|_2^2 \end{aligned} \tag{3}$$

The target of the generator is to produce samples that are of uniform distribution in B.

$$\begin{aligned} t_G(x) &= \frac{1}{|B|} \\ \ell_G(Z(x), y) &= t_G(x)\frac{1}{2}\|(z-\mu_y)\|_2^2 \\ &= \frac{1}{2|B_+|}\|(z-\mu_y)\|_2^2 + \frac{1}{2|B_-|}\|(z^{'} - \mu_y^{'})\|_2^2 \end{aligned} \tag{4}$$

Here $z^{'}$ and $\mu^{'}$ are the adversarial sample and mean respectively.

The generator structure in D-GAN consist of stacks of dilated convolutional neural network (CNN), each CNN layer applies a Batch Normalization [11] (BN) layer followed by ReLU activation layer. The discriminator on the other hand consists of multiple heads that use the same types of dilated convolutional network layers in a form of ensemble learning. The output independent heads are fused together before reaching the lower layer dilated convolutions. The discriminator structure ends with a 2-way fully connected layer.

3.3.1 Difference between the Softmax GAN and D-GAN

In the regular GANs, the goal of G is to fool D so that D cannot distinguish between generated or real samples,

Table 1: D-GAN architecture and hyperparameters

Operation	Kernel	Stride	Unit	BN	Non-Linearity
$G(z^{'})$					
dilated Convolutions	3×3	2×2	64	*yes*	ReLU
dilated Convolutions	3×3	2×2	64	*yes*	ReLU
dilated Convolutions	3×3	2×2	32	*no*	-
$D(z)$					
H_1					
dilated Convolutions	3×3	2×2	128	*yes*	ReLU
dilated Convolutions	3×3	2×2	80	*yes*	ReLU
dilated Convolutions	3×3	2×2	64	*yes*	ReLU
dilated Convolutions	3×3	2×2	32	*no*	-
H_2					
dilated Convolutions	5×5	3×3	256	*yes*	ReLU
dilated Convolutions	5×5	3×3	128	*yes*	ReLU
dilated Convolutions	5×5	3×3	80	*yes*	ReLU
dilated Convolutions	5×5	3×3	32	*no*	-
$D(z) = H_1 + H_2$					
Concatenate $D(z)$ and $G(z^{'})$					
$D(z, z^{'})$					
Dense			128	*no*	ReLU-
Dense			1	*no*	Linear
Optimizer	$\text{Adam}(\alpha = 10^{-5})$				
Batch size	64				
Epochs	100				

where the input of G is a random number sampled from a uniform distribution in the range [0,1]. In D-GAN, however, the generator G is trained to generate samples that is close to the real data under the newly learned metric. Since the generator plays the role of the adversarial malicious samples generation for the evasion attack, therefore, the inputs are real malicious data.

Another major difference between the Softmax GAN and D-GAN is the loss function. In D-GAN we replace the losses in Equtions (1) and (2), in Softmax GAN with the Mahalanobis distance loss, Equations (3) and (4), respectively.

4 Experiments and Results

In the experiment, Tensorflow [1] is adopted as the deep learning framework to implement D-GAN. The purposed model is run and evaluated on a Windows PC with Intel Quad Core Optiplex i7-2600 with 8G memory. D-GAN is trained with the 64 batch size for 100 epochs. We set both the generator and the discriminator learning rates to 0.0001, and the discriminator's weight clipping threshold to 0.01.

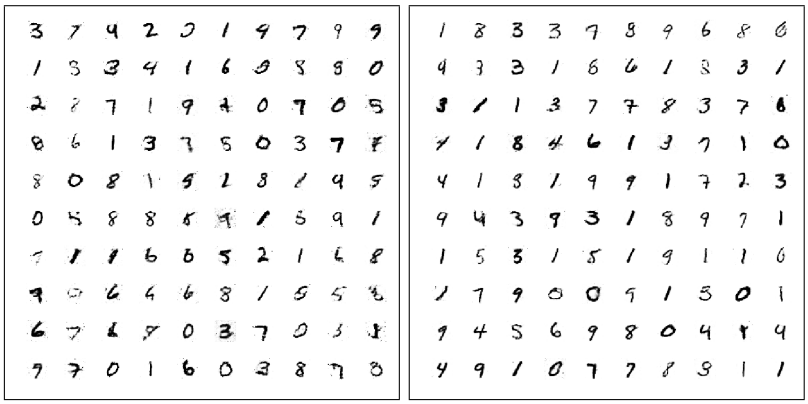
We train and test D-GAN shown in Table 1 on MNIST [13] and KDD99 [16]. For the KDD99 data, we follow the experimental setups of [26]. We evaluate D-GAN with the same metrics (Precision, Recall, F1 score) as the state-of-the art baselines. For the MNIST data, we convert the pixel value from $[0, 255]$ to $[0, 1]$ and evaluate models based on the stability of the discriminator to produce realistic images. The hyperparameters to train D-GAN on MNIST and KDD99 datasets are presented in Table 1.

4.1 D-GAN on MNIST Dataset

MNIST dataset is a database of handwritten digits. It consists of 28×28 pixels grayscale images with 10 output classes and having 60,000 examples as training set and 10,000 examples a test set. We compare the training stability of D-GAN to the Softmax loss. In this work we define training stability as the quality of images produced after each iteration. The results in Figure 2 shows that, D-GAN could generate more realistic images. In this regard, Mahalanobis distance loss indeed increase the stability of training GANs.

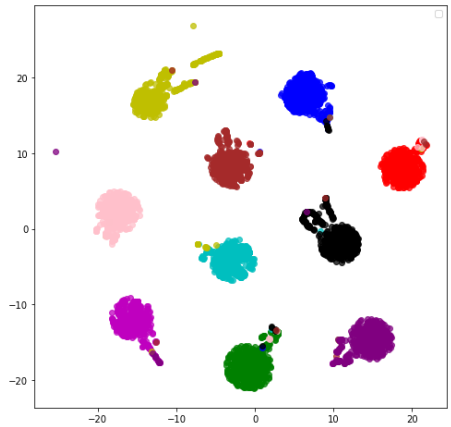
4.2 D-GAN on KDD Dataset

KDD dataset consists of a collection of network intrusion detection data used for The Third International Knowledge Discovery and Data Mining Tools Competition held in conjunction with KDD-99. We used 10 percent version of the KDDCUP99 data set. We randomly split the whole dataset into 50\50 for training and testing. We used data samples from the malicious class for training the discriminator, therefore because the proportion of data belonging

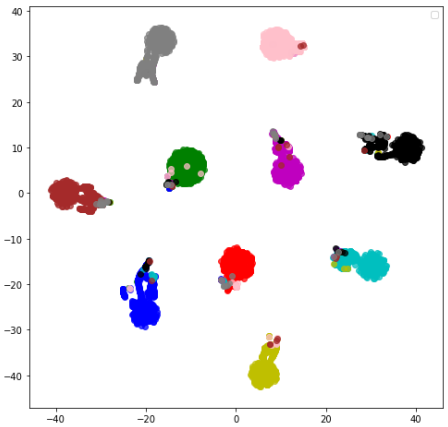


(a) Softmax GAN (b) D-GAN

Figure 2: Experimental results showing the stability of D-GAN. D-GAN can generate realistic images, less can be said about the Softmax GAN. Epochs = 100.



(a) MNIST training



(b) MNIST testing

Figure 3: The t-SNE representation of the deeply learned features in training set and testing set, via D-GAN dataset. The classes in the MNIST dataset are represent with the different colors. Epochs = 100.

to the malicious class is much larger than the proportion of data belonging to the normal class. We compare the D-GAN results on KDD99 dataset against several Softmax loss based GAN anomaly detection methods also trained on KDD99 dataset such as the work presented in [2, 15, 19, 23, 26, 27]. On the KDDCUP99 dataset, D-GAN can reach 95% F1 score with precision larger than 95% and recall higher than 96% better than that of the baselines as shown in Table 2.

Table 2: Classification accuracy on KDD99 dataset. Results for EFFICIENT-GAN, AnoGAN, MAD-GAN, ALAD, Fence-GAN, GANomaly were obtained from [26], [23], [15], [27], [19], [18] respectively while results for D-GAN are averages over 100 runs. High scores are indicated in bold.

Methods	Precision	Recall	F1
AnoGAN	0.88	0.83	0.89
EFFICIENT-GAN	0.92	0.95	0.93
GANomaly	0.83	0.84	0.84
MAD-GAN	0.94	0.96	0.94
ALAD	0.94	0.96	0.95
Fence-GAN	0.954	0.969	0.95
D-GAN	**0.961**	**0.972**	**0.969**

4.3 Exploration on the Distribution of Malicious Examples

To show that D-GAN discriminator can discriminate the learned features, we use t-SNE on MNIST dataset to illustrate the distribution in 2 dimensions. From Figure 3, we find that under the supervision of Mahalanobis distance loss, the deeply learned features are separable and also discriminative. Therefore we can conclude that using Mahalanobis distance loss increase the discriminative power of the GAN network which is crucial for anomaly detection.

5 Conclusion

Generative adversarial network (GAN) trained on Softmax cross entropy loss has been successfully extended to anomaly detection research fields as a detection classifier. However, in this paper, we identify that the Softmax cross entropy loss is less discriminative enough to classifying malicious samples. Inspired by this analysis, we propose the Mahalanobis distance loss as replacement of the SCE loss. Our model called D-GAN uses Mahalanobis distance loss to induce high-density regions in the feature space. In the high feature space, the real samples should be close to each other, while further away from the malicious samples, thus, minimizing the inter real samples distance and maximizing real and malicious samples distance, hence, increasing the discriminative power of GAN classifiers.

On the MNIST and KDD99 datasets, D-GAN outperforms existing methods in anomaly detection. We have shown that under the supervision of Mahalanobis distance loss, the deeply learned features are separable and also discriminative enough for anomaly detection.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, j. Levenberg, , D. Man, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vigas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," *Distributed, Parallel, and Cluster Computing*, 2015. arXiv:1603.04467.

[2] S. Akcay, A. A. Abarghouei, T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," *Computer Vision and Pattern Recognition*, 2018. arXiv:1805.06725.

[3] N. Carlini, D. A. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017. DOI: 10.1145/3128572.3140444.

[4] S. Chen, C. Gong, J. Yang, X. Li, Y. Wei, J. Li, "Adversarial metric learning," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018. (`https://www.ijcai.org/Proceedings/2018/0279.pdf`)

[5] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, A. C. Courville, "Adversarially learned inference," in *The 5th International Conference on Learning Representations (ICLR'17)*, 2017. arXiv:1606.00704.

[6] J. Donahue, P. Krähenbühl, T. Darrell, "Adversarial feature learning," in *The 5th International Conference on Learning Representations*, 2017. arXiv:1605.09782

[7] Y. Duan, J. Lu, W. Zheng, J. Zhou, "Deep adversarial metric learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. DOI: 10.1109/CVPR.2018.00294.

[8] F. Friedman, T. Hastie, R. Tibshirani, "The elements of statistical learning," *Springer Series in Statistics*, 2008. (`https://web.stanford.edu/~hastie/Papers/ESLII.pdf`)

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, 2014. (`https://papers.nips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf`)

[10] I. J. Goodfellow, J. Shlens, C. Szegedy, "Explaining and harnessing adversarial examples," in *The 3rd International Conference on Learning Representations (ICLR'15)*, 2015. arXiv:1412.6572.

[11] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015. arXiv:1502.03167.

[12] A. Kurakin, I. J. Goodfellow, S. Bengio, "Adversarial examples in the physical world," in *The 5th International Conference on Learning Representations (ICLR'17)*, 2016. arXiv:1607.02533.

[13] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[14] O. Ledoit, M. Wolf, "Improved estimation of the covariance matrix of stock returns with an application to portofolio selection," *Journal of Empirical Finance*, vol. 10, no. 5, pp. 603-621, 2003.

[15] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S. K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," *Machine Learning*, 2019. arXiv:1901.04997.

[16] M. Lichman, *KDD Cup 1999 Data*, 1999. (`http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`)

[17] C. Mao, Z. Zhong, J. Yang, C. Vondrick, B. Ray, "Metric learning for adversarial robustness," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, 2019. arXiv:1909.00900.

[18] F. D. Mattia, P. Galeone, M.D. Simoni, E. Ghelfi, "A survey on GANs for anomaly detection," *Machine Learning*, 2019. arXiv:1906.11632.

[19] C. P. Ngo, A. A. Winarto, C. K. L. Kou, S. Park, F. Akram, H. K. Lee, "Fence GAN: Towards better anomaly detection," *Machine Learning*, 2019. arXiv:1904.01209.

[20] T. Pang, C. Du, J. Zhu, "Max-mahalanobis linear discriminant analysis networks," in *Proceedings of the 35th International Conference on Machine Learning*, 2018. arXiv:1802.09308.

[21] T. Pang, K. Xu, Y. Dong, C. Du, N. Chen, J. Zhu, "Rethinking softmax cross-entropy loss for adversarial robustness," *Machine Learning*, 2019. arXiv:1905.10626.

[22] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy*, 2015. arXiv:1511.07528.

[23] T. Schlegl, P. Seebock, S. M. Waldstein, U. SchmidtErfurth, G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," *Information Processing in Medical Imaging*, 2017. arXiv:1703.05921.

[24] Y. Wen, K. Zhang, Z. Li, Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*, pp. 499-515, 2016.

[25] B. Wu, Z. L. Chen, J. Wang, H. M. Wu, "Scalable angular discriminative deep metric learning for face recognition," *Computer Vision and Pattern Recognition*, 2018. arXiv:1804.10899.

[26] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," *Machine Learning*, 2018. arXiv:1802.06222.

[27] H. Zenati, M. Romain, C. S. Foo, B. Lecouat, V. Chandrasekhar, "Adversarially learned anomaly detection," in *IEEE International Conference on Data Mining (ICDM'18)*, 2018. arXiv:1812.02288.

[28] W. Y. Zhong, T. Li, J. Chen, "Rethinking feature distribution for loss functions in image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. arXiv:1803.02988.

Biography

Benjamin Appiah is currently a Ph.D. candidate at University of Electronic Science and Technology of China, Chengdu, China. His research interests include machine learning and deep learning, data mining, big data analysis.

Zhiguang Qin is currently a Full Professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC), where he is also the Director of the Key Laboratory of New Computer Application Technology and the UESTC-IBM Technology Center. His research interests include medical image processing, computer networking, information security, cryptography, information management, intelligent traffic, electronic commerce, distribution, and middleware.

Obed Tettey Nartey is currently pursuing the Ph.D. degree. His research interests include artificial intelligence, computer vision, machine learning and pattern recognition, data mining, and deep learning.

Brighter Agemangcompleted his MSc Computer Science at UESTC in 2016. His current research interests are reinforcement learning, deep learning, and bioinformatics.

Ansuura JohnBosco Aristotle Kanpogninge is currently a Ph.D. candidate at University of Electronic Science and Technology of China, Chengdu, China. His research interests includes Cryptography, Block Chain, data mining, and big data analysis.

A Novel Reversible Data-Hiding Method Using Adaptive Rhombus Prediction and Pixel Selection

Thai-Son Nguyen

(Corresponding author: Thai-Son Nguyen)

School of Engineering and Technology, TraVinh University, Vietnam

(Email: corresponding_thaison@tvu.edu.vn)

(Received July 22, 2020; Revised and Accepted Dec. 19, 2020; First Online June 7, 2021)

Abstract

The reversible data hiding (RDH) technique is used to conceal secret data within images such that the original image can be fully recovered upon the extraction of hidden data. Such a technique has attracted much interest from researchers in different fields. In this article, to improve embedding capacity further while maintaining minimum distortion of stego images, a novel reversible data-hiding algorithm is proposed by using the combination of adaptive rhombus prediction and pixel selection. Furthermore, to preserve the image quality well and increase the embedding capacity significantly, the proposed scheme has carefully determined a suitable pixel set and image region for embedding data. The experimental result shows that our method brings better performance than some previous schemes in virtual image quality and embedding capacity.

Keywords: Histogram Shifting; Multiple-Layer Embedding; RDH; Reversibility; Rhombus Prediction

1 Introduction

Reversible data hiding (RDH) is an algorithm that aims to extract embedded secret data exactly and to reconstruct the cover image to its original version without any distortion. Due to these properties, several RDH techniques have been proposed in various fields, such as video error-concealment coding, the military, and medical images [2, 9, 15, 22, 23]. In general, the performance of the RDH technique is evaluated based on the behavior of embedding capacity (EC) and virtual image quality. For a given EC, embedding distortion is maintained as small as possible to achieve good image quality of stego images.

In the last few decades, many RDH schemes that have been proposed can be divided into four types: lossless compression [1, 3, 4, 10, 26], difference expansion (DE) [6, 8, 12, 19, 20, 27, 28], histogram shifting (HS) [5, 7, 11, 13, 14, 16, 18, 21, 25], and integer transform [17, 24]. Among these RDH schemes, HS-based RDH algorithms have attracted much attention of researchers due to good virtual image quality and sufficient EC. The HS-based RDH method was first introduced by Ni *et al.* [16]. In this scheme, Ni et al generated a histogram of the cover image, then they selected peak points and zero points to conceal the secret data. In their scheme, each pixel is altered by no more than one value. Therefore, Ni *et al.*'s scheme achieved a high quality of stego-images; however, its EC is not satisfied when the average EC is always smaller than 7,000 bits. To improve EC further, Hong *et al.* [7] used prediction errors for hiding secret data. In the scheme [7], the high redundancy of pixels is exploited for containing the secret bits. Consequently, their performance is superior to those of DE-based and previous HS-based schemes [16, 19, 20].

In [18], Sachnev *et al.* used rhombus prediction to classify image pixels into two sets – Dot and Cross sets – for HS-based RDH scheme. Their scheme obtained high EC while guaranteeing good image quality. A similar scheme was introduced in [24] with optimal side information selection. Later, a new RDH algorithm, based on the interpolation mechanism, is proposed by Luo *et al.* [14]. In this scheme, interpolation errors are calculated as the difference between the pixel value and the corresponding interpolation value. Then, these difference values are used for hiding the secret message. To guarantee high image quality, in [14], all cover pixels are only altered slightly in the embedding steps; however, the image is distorted significantly when the large EC is embedded. To increase EC while guaranteeing good image quality, in [11], Li et al designed two HS-based schemes by exploiting a nine-dimensional histogram. In their scheme, suitable pixels are selected carefully. Then, two functions, shifting and embedding, are used to generate the room for containing the secret bits. Li *et al.*'s performance was superior to those of previous RDH schemes [11, 12, 14, 18]. Recently, Wang *et al.* [25] proposed a new multiple HS-based RDH technique by employing a genetic algorithm (GA). To maintain reversibility, Wang *et al.* utilized prediction

errors of each pixel to embed secret bit. However, it also causes significant distortion on stego images, if the large EC is embedded.

In this article, a new HS-based RDH method is proposed by combining adaptive rhombus prediction and pixel selection. Inspired by pixel selection techniques proposed in previous schemes [11, 14, 18], for each pixel in this scheme, the local variance value is computed to determine whether it is suitable to embed the secret bit or not. To compute the prediction errors, in the proposed method, an adaptive rhombus prediction algorithm is applied. In addition, to keep the embedding distortion small and the EC high, the optimal information is selected and used during the embedding process. Experimental results demonstrate that the proposed method generates better performance than some previous schemes in terms of EC and image quality.

The remainder of this paper is structured as follows. The brief of Li *et al.*'s method [11] is reviewed in Section 2. Then, Section 3 gives the detail of the proposed method. In Section 4, our proposed method' performance is analyzed in compared with some existing schemes. Eventually, conclusion is provided in Section 5.

2 Related Work

In 2013, Li *et al.* [11] explored a nine-dimensional histogram for hiding the secret message. Take for example, the image block Z sized of 3×3 as shown in Figure 1.

For the first pixel Z_1, the two neighboring pixels Z_2 and Z_4 are used to predict its value by using Equation (1):

$$Z_i' = \left\{ \begin{array}{l} k, \text{ if } Z_1 \geq k \\ l-1, \text{ if } Z_1 < l \end{array} \right\} \quad (1)$$

where $k = max(Z_2, Z_4)$, $l = min(Z_2, Z_4)$, and $k \geq l$.

For embedding the secret data, the local complexity $Com(Z)$ of Z is calculated as Equation (2):

$$Com(Z) = max(Z_2, Z_3, ..., Z_9) - min(Z_2, Z_3, ..., Z_9). \quad (2)$$

After obtaining the local complexity of Z, this local complexity value is used to select the embeddable blocks. To embed a bit b into the block, the $Embed()$ function is used as defined in Equation (3):

$$Embed_b(Z) = \left\{ \begin{array}{ll} (Z_1 + b, Z_2, Z_3, ..., Z_9), & \text{if } Z_1 - k = 0, Com(Z) < T \\ (Z_1 - b, Z_2, Z_3, ..., Z_9), & \text{if } Z_1 - l = -1, Com(Z) < T \\ (Z_1 + 1, Z_2, Z_3, ..., Z_9), & \text{if } Z_1 > k, Com(Z) < T \\ (Z_1 - 1, Z_2, Z_3, ..., Z_9), & \text{if } Z_1 > l-1, Com(Z) < T \\ Z & \text{otherwise} \end{array} \right\} \quad (3)$$

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Figure 1: Image block sized of 3×3

where T is the smallest positive integer threshold that is a selected to make sure all of the secret bits to be embedded. Notice that this scheme [11] used the local complexity estimator for selecting embeddable areas in the cover image to achieve a more concentrated histogram. With the local complexity estimator, only blocks with the value of local complexity (smaller than T) will be used to contain the secret bits; however, only the pixel Z_1 of each block sized of 3×3 is used to contain one secret bit, resulting in a limited amount of embedding spaces. Therefore, this scheme is not favorable to hide a large amount of secret data. Obviously, there is a high correlation of the pixels in natural images; therefore, the HS-based RDH schemes seek to achieve high EC and to preserve high-quality stego images; however, these existing schemes are still limited in their performance when the average image quality is smaller than 59 dB for embedding 10,000 secret bits. To overcome these shortcomings, a new RDH scheme is proposed by using a combination of adaptive rhombus prediction and pixel selection, and the detail of which is presented in the next section.

3 Proposed Method

In this section, a novel RDH method is introduced. The flowchart presenting this method is present in Figure 2. The embedding process is briefly described as follows. First, the cover image is separated into two sub-images, *i.e.* A and B. After that, in the sub-image A, all of the pixels is evaluated to classify into two sets, Cross and Dot sets, as can be seen in Figure 3. Then, each set is divided into two parts, *i.e.*, Part 1 and Part 2, consisting of smooth and complex pixels, respectively. To hide the secret data into the Cross set or Dot set, the prediction error histogram of Part 1 is generated. According to the size of the embedded secret data, we optimally select the pair of peak and zero points. Finally, the secret bit is hidden by shifting the prediction error histogram of Part 1. It is noted that pixels in Part 1 are modified by no more than one value, and the Dot set is applied for predicting

pixels of the Cross set, and vice versa. By so doing, the cover image can be prevented from significant embedding distortion. The proposed method can be divided into two subsections, *i.e.* embedding algorithm and extracting algorithm.

3.1 Embedding Algorithm

Assuming that the cover image I with a size of $W \times H$, the secret data $S = s_1, s_2, \ldots s_L$, where $s_i = \{0, 1\}$, and L is the length of S. For embedding data, the image I is divided into sub-image A and sub-image B, as shown in Figure 3. Then, the proposed embedding algorithm is presented as followings.

Step 1 (Pixel partition): Partition the sub-image A into two sets, *i.e.*, Cross (X) and Dot (O) sets, (see Figure 3(a)). After that, the LSBs of pixels in sub-image B is recorded into the bit stream LSB_B. Then, the secret data S is generated by concatenating the bit stream LSB_B into the secret message M.

Step 2 (Pixel selection): To avoid the complexity region, the smooth pixels are selected and used for RDH to decrease embedding distortion as much as possible. The local complexity of each pixel P_X in the Cross set is calculated by using its local-variance (LV) value, defined in Equation (4). Then, the threshold TH is used to determine whether the smooth pixels or not. Equation:

$$LV_P X = dv + dh. \tag{4}$$

where $dh = |P_O^1 - P_O^3|$ is the horizontal variance, $dv = |P_O^2 - P_O^4|$ is the vertical variance, and P_O^i is four adjacent Dot pixels of P_X (see Figure 3(b)). According to the values of the local variance and the selected threshold TH, the pixel P_X is then classified into Part 1 or Part 2 as follows:

- Part 1 (smooth pixels): $P_X \in X : LV_X < TH$.
- Part 2 (complex pixels): $P_X \in X : LV_X \geq TH$.

Step 3 (Computation of prediction errors): For each pixel of Part 1 in the Cross set, according to the values of dv and dh, four adjacent Dot pixels are adaptively used to predict the value of P_X as follows.

$$P'_X = \left\{ \begin{array}{ll} round(\frac{P_O^1 + P_O^3}{2}), & \text{if } dv > dh \\ round(\frac{P_O^2 + P_O^4}{2}), & \text{if } dv \leq dh. \end{array} \right\} \tag{5}$$

Then, calculate the corresponding prediction error EX by Equation (6).

$$E_X = P_X + P'_X. \tag{6}$$

Step 4 (Determination of optimal pair of peak and zero points): As can be seen in existing HS-based RDH techniques, the peak point and the zero point are determined in the prediction error histogram for embedding data. Thus, the embedding capacity C is considered as the frequency of the peak point: $C = F(Peak)$, where $F(.)$ is the frequency function of the histogram. Different from existing HS-base techniques, after obtaining all prediction errors E_X of Part 1 in the Cross set, the proposed method optimally selects pair of peak and zero points in the histogram of E_X for embedding the secret data S as following.

- From the value of 0, scan negative and positive axes of the histogram, to find the first two matched zero points Z_l and Z_r, as shown in Figure 4(a).
- From the point Z_l toward the value of 0, search for a suitable peak point P_l, such that the constraint, $F(P_l) \geq L$, is satisfied.
- Similarly, from the point Z_r toward the value of 0, search for a suitable point P_r, such that the constraint, $F(P_r) \geq L$, is satisfied.
- Among two candidate pairs of peak and zero points, *i.e.*, (P_l, Z_l) and (P_r, Z_r), the optimal one, denoted as (P^*, Z^*), is selected with the smaller shifting distortion.

Step 5 (Embedding process and generation of the stego image): After determining the optimal pair (P^*, Z^*), for hiding the secret data S, all bins between P^* and Z^* are shifted to the Z^* direction to generate room bin near the peak point P^*. Assume that $P^* < Z^*$. Then, each prediction error E_X is processed by either shifting its value if $E_X \in (P^*, Z^*)$ or embedding one secret bit s_i, if $E_X = P^*$, which is defined in Equation (7).

$$E'_X = \left\{ \begin{array}{ll} E_X + 1, & \text{if } E_X \in (P^*, Z^*) \\ E_X + s_i, & \text{if } E_X = P^*. \\ E_X, & \text{otherwise} \end{array} \right\} \tag{7}$$

After that, the stego pixels of Part 1 in the Cross set are computed by using Equation (8).

$$P''_X = P'_X + E'_X. \tag{8}$$

According to the values of stego pixels P''_X in the Cross set, the same steps are used to embed the secret data into pixels of the Dot set to generate stego pixels P''_O. Finally, P''_X and P''_O are combined to generate the stego image.

Instead of directly using the pair of peak and zero points as was done in the existing HS-based techniques, two candidate pairs of peak and zero points are determined, and the optimal one is used to hide secret data. Here, for single-layer embedding, assume that $F(P_l) \geq L$ and $F(P_r) \geq L$; however, if the length of the secret data $L > F(P_l)$ and $L > F(P_r)$, multi-layer embedding with

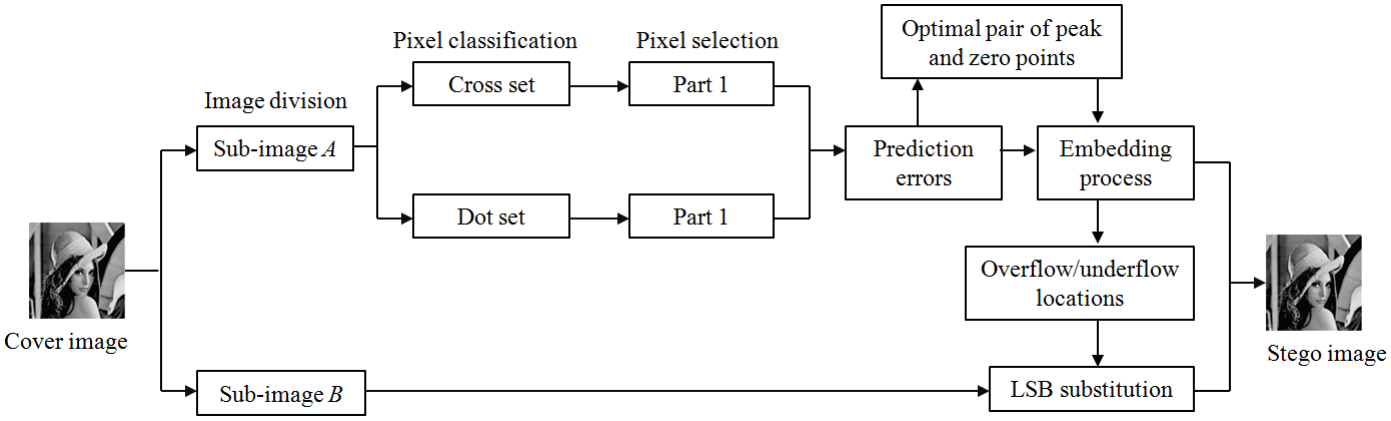


Figure 2: Flowchart of the proposed embedding process

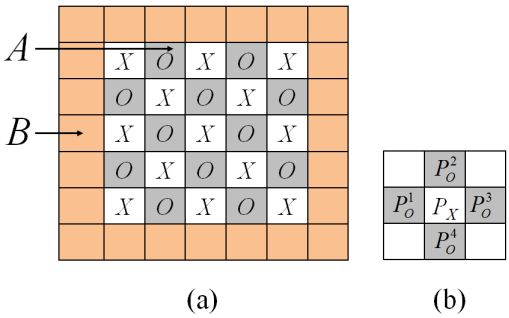


Figure 3: Division of the cover image

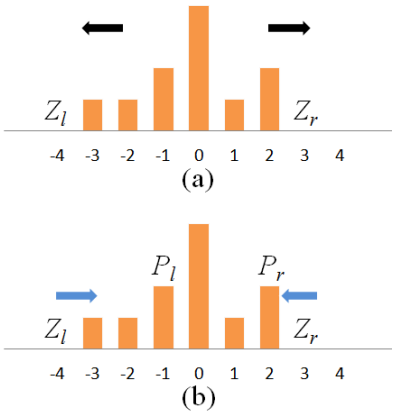


Figure 4: Division of the cover image

two steps is applied. Step 1: Search for two highest frequency points P_l and P_r on the negative and positive axes of the histogram, if $L > F(P_l)$ and $L > F(P_r)$, then select the higher frequency P as $P = maxP_l, P_r$, and the nearest zero point Z of such peak point P is selected for embedding the secret data S. After that, calculate $L = L–F(P)$, and the Step 1 is repeated for the next embedding layer if $L > F(P_l)$ and $L > F(P_r)$. Otherwise, Step 2, the optimal single-layer embedding process, is used. Step 2 that is discussed above is always applied for the final-layer embedding.

Note that all of the smooth pixels (Part 1) in both the Cross and Dot sets are chosen to conceal the secret data, and these pixels are modified by no more than one value. As a result, few pixels may cause an overflow/underflow of the pixels. Therefore, instead of using the location map as was done in [18, 24], that is time-consuming and distorts the cover image significantly since the large size of location map is also embedded into the cover image because of the reversibility reason. The proposed method only records those overflow/underflow locations into the binary sequence BS. In the proposed method, not only selecting the optimal pair of peak and zero points, to minimize the embedding distortion, but threshold TH is also taken as the smallest one such that the proposed method has the ability to embed the required capacity. Then, the sub-image B is used to contain extra information, *i.e.*, the binary sequence BS, the optimal pair (P^*, Z^*), and the threshold TH by LSB substitution. It is noted that the size of sub-image B should be contained the extra information completely. Table 1 shows that our scheme with different selected thresholds TH for single-layer embedding achieves better than previous schemes since the proposed method achieves the greater virtual quality for different images. It is obvious that our method shows the effectiveness of a combination of adaptive rhombus prediction and the pixel selection strategy.

Table 1: Performance comparison of the proposed method with different selected threshold TH for single-layer embedding

Image	Criteria	TH = 5	TH = 10	TH = 15	TH = 20	[18]	[14]	[11]	[24]
Lena	Pure EC	0.159	0.246	0.280	0.296	0.138	0.236	0.114	0.150
Lena	PSNR	57.42	54.65	52.02	51.77	52.60	48.12	53.62	52.51
Baboon	Pure EC	0.009	0.026	0.041	0.051	0.030	0.040	0.039	0.033
Baboon	PSNR	64.62	60.01	56.80	55.87	49.58	47.88	50.03	48.62
Boat	Pure EC	0.220	0.257	0.293	0.274	0.080	0.149	0.109	0.078
Boat	PSNR	56.41	54.88	52.28	53.77	52.40	47.36	52.22	53.52
Sailboat	Pure EC	0.112	0.152	0.206	0.241	0.083	0.140	0.101	0.072
Sailboat	PSNR	52.64	48.12	46.04	45.78	52.18	47.25	52.89	52.87
Peppers	Pure EC	0.131	0.230	0.254	0.268	0.080	0.151	0.100	0.075
Peppers	PSNR	57.61	52.73	53.57	53.11	53.12	47.32	53.60	53.59
Airplane	Pure EC	0.246	0.315	0.332	0.340	0.249	0.299	0.179	0.199
Airplane	PSNR	55.94	53.69	51.82	51.52	51.33	47.14	55.63	51.32

3.2 Extracting Algorithm

The extracting algorithm contains five steps. Firstly, the stego image is divided into sub-image A and sub-image B. Later on, LSB bits are extracted from the sub-image B, to determine whether pixels contained the secret data or not. The threshold TH and the optimal pair of peak and zero points are also read from LSB bits in the sub-image B. According to the embedding algorithm, after obtaining the values of stego pixels P''_X in the Cross set, the same steps are used to embed the secret data into pixels of the Dot set to generate stego pixels P''_O. Therefore, to maintain the reversibility, in the extracting algorithm, the secret data are extracted from pixels of the Dot set P''_O and the Cross set P''_X, respectively. The extracting algorithm is used for reconstructing the secret bits S and restoring to the original version of cover images as following.

Step 1 (Pixel Partition): Separate the stego image into two sub-images, *i.e.*, A and B. Then, from B, read LSBs to determine overflow/underflow locations BS, TH, and (P^*, Z^*). Partition the sub-image Ainto two sets, Cross (X) and Dot (O) sets.

Step 2 (Pixel selection): For each stego pixel P''_O in the Dot set, and $P''_O \notin X$, re-calculate its local-variance (LV) using Equation (4). Then, according to the local variance and the threshold TH, pixels P''_O are classified into Part 1 or Part 2.

- Part 1 (smooth pixels): $\{P''_O \in O : LV_O < TH$, and $P''_O \notin X\}$.
- Part 2 (complex pixels): $\{P''_O \in O : P''_O \notin Part1\}$.

Step 3 (Computation of prediction errors): Compute the predicted value P'_O of the current stego-pixel P''_O by Equation (5), and determine the corresponding stego prediction error E'_O in the Dot set as

$$E'_X = P''_O - P'_O. \tag{9}$$

Step 4 (Extracting process): Based on the extracted optimal pair (P^*, Z^*), the secret bit si is extracted by Equation (10), and Equation (11) is used to restore the prediction error.

$$s_i = \left\{ \begin{array}{ll} 0, & \text{if } E'_O = P^* \\ 1, & \text{if } E'_O = P^* + 1. \end{array} \right\} \tag{10}$$

$$E_O = \left\{ \begin{array}{ll} E'_O - 1, & \text{if } E'_O \in (P^*, Z^*] \\ E'_O, & \text{otherwise} \end{array} \right\} \tag{11}$$

Then, pixels of Part 1 in the Dot set are reconstructed as

$$P_O = P'_O + E_O. \tag{12}$$

Step 5 (Restoration of the original image): According to the values of reconstructed pixels P_O in the Dot set, same steps are used to extract the secret data from pixels of the Cross set and reconstruct the original pixels P_X in the Cross set. Then, the cover image I is restored by combining of P_X and P_O. Once the secret data S is completely extracted, it is divided into the secret message M and the bit stream LSB_B. Then, LSBs of pixels in B is replaced with LSB_B to recover the original image I. It is noted that, when these five steps are performed completely, the secret message M is extracted correctly and the original image I is restored precisely.

4 Experimental Results

In this section, all results of the proposed method are analyzed in comparison with four previous schemes [11, 14, 24, 25]. Six grayscale images with a size of 512×512 in Figure 5, were used in the experiment. It is noted that two different schemes are introduced in [11]. Their scheme 1 provided the higher EC, whereas the better image quality was obtained by their scheme 2 when the small EC was embedded. To make a fair comparison, only

the best results of each scheme were used for comparison.Figure 6shows the comparison results of our scheme and four other schemes. Here, we vary the embedding rate (ER) from 0.1 bpp to its maximum with the step size 0.1. Figure 6 shows that, for all of the six tested images, the proposed method achieved superior performance to those of four previous schemes [11, 14, 24, 25]. The proposed method yielded a very good image quality with high EC. Lena image can be taken as an example. When the embedding rate is 0.6 bpp, the proposed method still maintains the good image quality (larger than 45 dB). The performance of Wang *et al.*'s scheme [25] is superior to those of three schemes [11,14,24] when embedding the secret data of small sizes. This is because, in the scheme [25], smooth pixels were also selected based on the GA algorithm. However, our scheme still outperforms Wang *et al.*'s scheme [25]. In particular, the proposed method produced the greater virtual image quality than the scheme [25], when average gains are 2.94 dB and 2.39 dB for an EC of 10,000 bits, 20,000 bits, respectively, with single-layer embedding, as can be seen in Tables 2 and 3. Although the proposed method and two other schemes [24,25] are based on both histogram shifting and rhombus prediction techniques, however, the better performance was obtained by the proposed method than by two other schemes [24,25] in most cases. In the proposed method, the threshold TH is used to control selected pixels, which guarantees that only pixels in the smooth region are embedded into the secret bits. Moreover, the adaptive rhombus prediction technique is utilized to reduce the embedding distortion in complex regions. In addition, instead of using the pair of peak and zero points as in existing HS-based RDH schemes, our method selected the most suitable one of the peak and zero points for embedding data, which is another reason for the superior of the proposed method (see in Figure 6).

In addition, the performance obtained from the proposed method is compared with six previous schemes [6, 11, 14, 24, 25, 27] at low embedding capacity when single-layer embedding is applied. Tables 2 and 3 show the comparison results for EC of 10,000 bits and 20,000 bits, respectively. From Tables 2 and 3, it is obvious that the proposed method shows better image quality than other existing schemes. Table 4 shows that the optimal pairs and sets are used for embedding with different sizes of the secret data. It is noted that the proposed method used different pairs of peak and zero points to minimize embedding distortion when the different sizes of secret data are embedded.

5 Conclusions

In this article, a novel HS-based RDH method based on the combination of adaptive rhombus prediction algorithm and pixel selection technique is proposed. To increase the EC while guaranteeing the small distortion of stego images, the local variance value of pixels in the cover image is calculated to determine embeddable pixels. Moreover, adaptive rhombus prediction algorithm is used for calculating the prediction errors. Then, optimal information is used in the proposed embedding phase to guarantee high performance. Experimental results suggested that the performance of our proposed method is improved further in comparison with previously rhombus prediction HS-based RDH techniques in terms of the EC and the image quality.

Acknowledgments

This study was supported by the Tra Vinh University under grant 207/HD-HDKH-DHTV. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] C. C. Chang and T. S. Nguyen, "A reversible data hiding scheme for SMVQ indices," *Informatica*, vol. 25, no. 4, pp. 523–540, 2014.

[2] C. C. Chang, T. S. Nguyen, and C. C. Lin, "A virtual primary key for reversible watermarking textual relational databases," in *Intelligent Systems and Applications*, pp. 756–769, Dec. 2014.

[3] C. C. Chang, T. S. Nguyen, and C. C. Lin, "A reversible compression code hiding using soc and SMVQ indices," *Information Sciences*, vol. 300, no. 10, pp. 85–99, 2015.

[4] W. J. Chen and W. T. Huang, "Vq indices compression and information hiding using hybrid lossless index coding," *Digital Signal Processing*, vol. 19, no. 3, pp. 433–443, 2009.

[5] G. Y. Gao, S. K. Tong, Z. H. Xia, B. Wu, L. Xu, and Z. Q. Zhao, "Reversible data hiding with automatic contrast enhancement for medical images," *Signal Processing*, vol. 178, 2021.

[6] W. He, K. Zhou, J. Cai, L. Wang, and G. Xiong, "Reversible data hiding using multi-pass pixel value ordering and prediction-error expansion," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 351–360, 2017.

[7] W. Hong, T. S. Chen, and C. W. Shiu, "Reversible data hiding for high quality images using modification of prediction errors," *Journal of Systems and Software*, vol. 82, no. 11, pp. 1833–1842, 2009.

[8] Y. Hu, H. K. Lee, and J. Li, "De-based reversible data hiding with improved overflow location map," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 250–260, 2009.

[9] R. Kumar and K. H. Jung, "Robust reversible data hiding scheme based on two-layer embedding strategy," *Information Sciences*, vol. 512, pp. 96–107, 2020.

[10] J. D. Lee, Y. H. Chiou, and J. M. Guo, "Reversible data hiding based on histogram modification

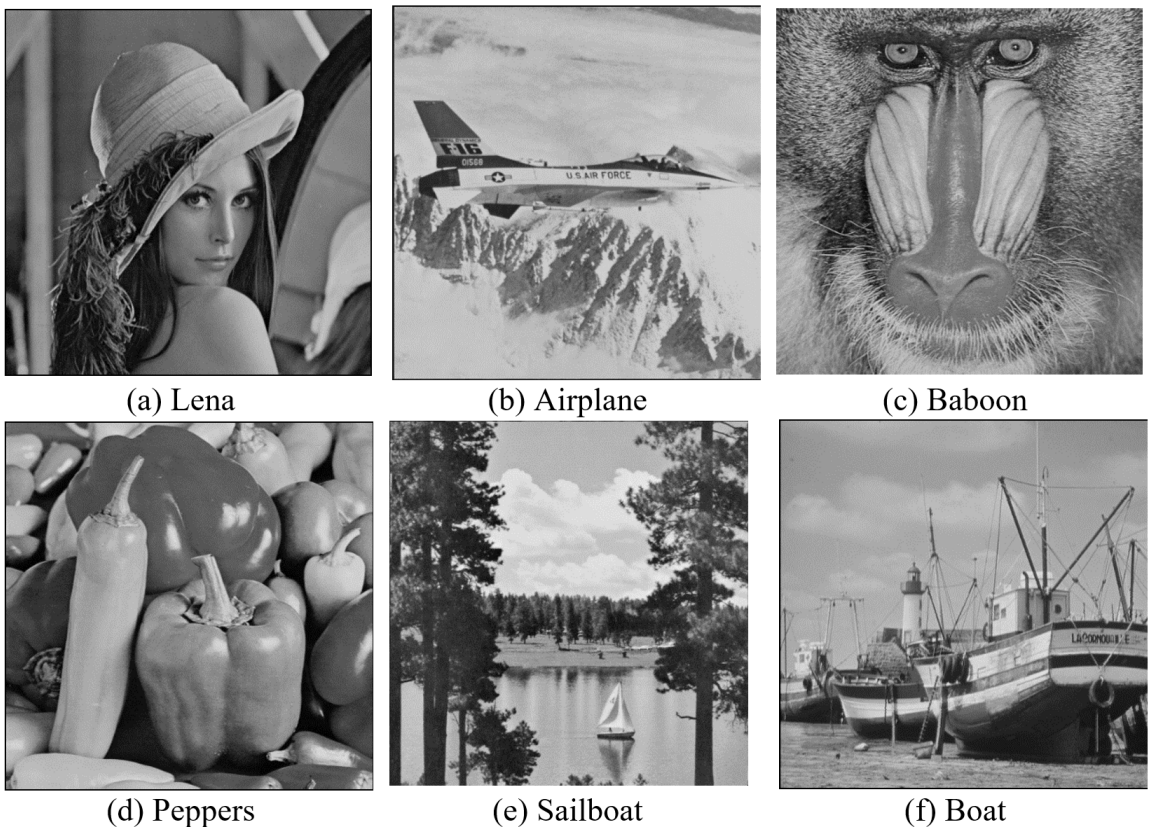


Figure 5: Six tested grayscale images sized 512x512

Table 2: Comparison of the image quality (dB) of five RDH schemes for EC of 10,000 bits with single-layer embedding

Images	**Proposed**	**Xiao [27]**	**Wang [25]**	**He [6]**	**Li [11]**	**Luo [14]**	**Wang [24]**
Lena	62.37	60.92	60.14	60.64	59.37	57.35	57.94
Airplane	63.42	63.97	61.93	63.45	62.65	57.97	59.25
Baboon	56.53	56.23	55.22	54.01	54.41	51.08	52.61
Peppers	62.40	58.76	58.04	59.29	56.89	55.23	56.64
Sailboat	61.68	59.87	57.45	59.71	58.27	55.74	57.38
Boat	61.47	58.34	57.47	58.28	57.16	54.07	56.31
Average	61.31	59.68	58.38	59.23	58.13	55.24	56.69
Gain of PSNR	-	1.63	2.94	2.08	3.19	6.07	4.62

Table 3: Comparison of the image quality (dB) of five RDH schemes for EC of 20,000 bits with single-layer embedding

Images	**Proposed**	**Xiao [27]**	**Wang [25]**	**He [6]**	**Li [11]**	**Luo [14]**	**Wang [24]**
Lena	59.64	57.32	56.81	56.81	55.93	53.85	55.87
Airplane	59.28	60.47	59.59	59.59	59.26	55.44	57.31
Peppers	58.36	54.99	55.11	55.11	53.31	52.26	53.71
Sailboat	58.47	54.58	54.53	54.53	53.19	52.17	54.58
Boat	56.31	54.13	54.07	54.07	53.05	51.19	53.36
Average	58.41	56.30	56.02	56.02	54.95	52.98	54.97
Gain of PSNR	-	2.11	2.39	2.39	3.46	5.43	3.45

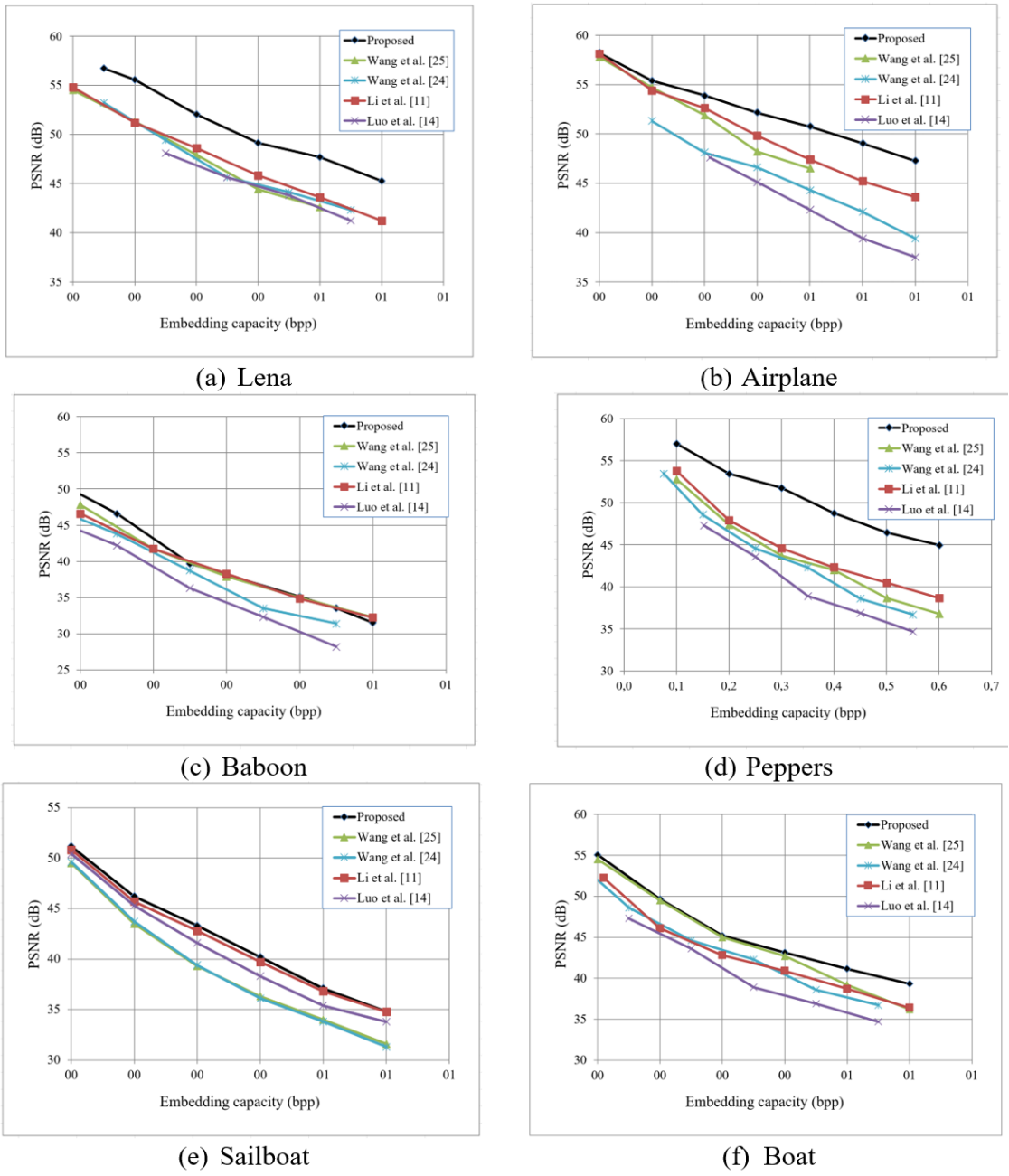


Figure 6: Performance comparison of the proposed method and four previous schemes [11, 14, 24, 25]

Table 4: Parameters used by the proposed method with the threshold TH = 5 for Lena image at low EC

ER (bpp)	Embedding set	Optimal peak and zero points	PSNR (dB)
0.035	Cross	1,20	62.87
0.035	Dot	(-1,-18)	62.31
0.09	Cross	0,20	59.12
0.09	Dot	(0,-18)	59.14
0.12	Cross	0,20	59.11
0.12	Dot	(0,-18)	57.72

of SMVQ indices," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 638–648, 2010.

[11] X. Li, B. Li, B. Yang, and T. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2181–2191, 2013.

[12] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524–3533, 2011.

[13] M. Long, Y. Zhao, X. Zhang, and F. Peng, "A separable reversible data hiding scheme for encrypted images based on tromino scrambling and adaptive pixel value ordering," *Signal Processing*, vol. 176, 2020.

[14] L. Luo, Z. Chen, M. Chen, X. Zeng, , and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2010.

[15] D. C. Nguyen, T. S. Nguyen, F. R. Hsu, and H. Y. Hsien, "A novel steganography scheme for video H. 264/AVC without distortion drift," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 16033–16052, 2019.

[16] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[17] F. Peng, X. Li, and B. Yang, "Adaptive reversible data hiding scheme based on integer transform," *Signal Processing*, vol. 92, no. 1, pp. 54–62, 2012.

[18] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989–999, 2009.

[19] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, pp. 721–730, 2007.

[20] J. Tian, "Reversible data hiding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 890–896, 2003.

[21] P. H. Vo, T. S. Nguyen, V. T. Huynh, and T. N. Do, "A novel reversible data hiding scheme with two-dimensional histogram shifting mechanism," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28777–28797, 2018.

[22] P. H. Vo, T. S. Nguyen, V. T. Huynh, T. C. Vo, and T. N. Do, "Secure and robust watermarking scheme in frequency domain using chaotic logistic map encoding," in *Advanced Computational Methods for Knowledge Engineering*, pp. 346–357, Dec. 2019.

[23] P. H. Vo, T. S. Nguyen, V. T. Huynh, T. C. Vo, and T. N. Do, "A high-capacity invertible steganography method for stereo image," *Digital Media Steganography*, pp. 99–122, 2020.

[24] J. Wang, J. Ni, and Y. Hu, "An efficient reversible data hiding scheme using prediction and optimal side information selection," *Journal of Visual Communication and Image Representation*, vol. 25, pp. 1425–1431, 2014.

[25] J. Wang, J. Ni, X. Zhang, and Y. Q. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 315–326, 2017.

[26] J. X. Wang and Z. M. Lu, "A path optional lossless data hiding scheme based on vq joint neighboring coding," *Information Sciences*, vol. 179, no. 19, pp. 3332–3348, 2009.

[27] M. Xiao, X. Li, Y. Wang, Y. Zhao, and R. Ni, "Reversible data hiding based on pairwise embedding and optimal expansion path," *Signal Processing*, vol. 158, pp. 210–218, 2019.

[28] J. Zhou and O. C. Au, "Determining the capacity parameters in pee-based reversible image watermarking," *IEEE Signal Processing Letters*, vol. 19, no. 5, pp. 287–290, 2012.

Biography

Thai-Son Nguyen received the M.S. and Ph.D. degrees from Feng Chia University, Taichung, Taiwan, in 2011 and 2015, respectively, all in computer science. he served as a lecturer in Tra Vinh University,Vietna, from 2006. From 2019, he was Dean of the school of Engineering and Technology, Tra Vinh University. He is currently an Associate professor. His research interests include image processing, information hiding, image recognition, information security, and artificial intelligence applications.

Analysis of Two Secure Three-Party Computation Protocols for Triangle Area

Lihua Liu[1] and Jie Cao[2]
(Corresponding author: Lihua Liu)
Department of Mathematics, Shanghai Maritime University[1]
Haigang Ave 1550, Shanghai, 201306, China
School of Computer Science and Technology, Soochow University, China[2]
Email: liulh@shmtu.edu.cn
(Received Jan. 15, 2020; Revised and Accepted Dec. 10, 2020; First Online June 12, 2021)

Abstract

Secure three-party computation is a special instance of the general secure multi-party computation, which is always hard to deal with. In this note, we show that the two secure three-party computation protocols [Int. J. Inf. Sec., 15 (2016), 1-13] are flawed. The schemes are just based on common public-key encryption and a general linear transformation. No sophisticated technique, such as zero-knowledge proof, oblivious transfer, cut-and-exchange, garbled circuit, and homomorphic encryption, is integrated into it. We find any two parties can conspire to retrieve the other party's input by solving normal quadratic equations or normal linear equations.

Keywords: Collusion Attack; Heron's Formula; Secure Multi-Party Computation; Shoelace Formula

1 Introduction

The primitive of secure multi-party computation (MPC) introduced by Yao [24] has many applications such as e-voting, e-auction [8], and e-cash [21]. It is usual that a MPC protocol has to make use of some sophisticated cryptographic tools, such as oblivious transfer [1,3,4,11, 13], zero-knowledge proof [8,9,17], cut-and-exchange [12, 15,26], shuffle [5,14,23], garbled circuit [18,19,22,25], and homomorphic encryption [2,6,7,10,20].

Very recently, Liu *et al.* [16] have presented two secure three-party computation protocols for triangle area. Both two protocols require that each participator's input can not be recovered by any adversary, while all participators know the final common output. But we have noticed that none of sophisticated cryptographic tools had been integrated into the two protocols. The schemes only use a common public key encryption and a linear transformation. Intuitively, the protocols are prone to failure. In this note, we show that both two protocols are insecure against the general collusion attack.

2 Analysis of Protocol-1

2.1 Review of Protocol-1

In the scheme [16], there are three parties, A, B, C, and each has a point $P_i = (x_i, y_i) \in \mathbb{Z}^* \times \mathbb{Z}^*$, $i \in \{a, b, c\}$. Let $G(\cdot)$ be a pseudorandom generator, and $r_a, r_{a1}, r_b, r_{b1}, r_c, r_{c1}$ be the seeds chosen by the three parties, respectively. E_a is the length of the opposite edge of vertex (x_a, y_a). So are E_b and E_c. $\mathrm{Enc}_{PK_c}(\cdot)$ denotes the encryption algorithm corresponding to the party C's public key PK_c. So are $\mathrm{Enc}_{PK_a}(\cdot)$ and $\mathrm{Enc}_{PK_b}(\cdot)$. The Heron's formula says that the area of a triangle with edges E_a, E_b, E_c is

$$S_{area} = \sqrt{P \cdot (P - E_a) \cdot (P - E_b) \cdot (P - E_c)},$$

where $P = (E_a + E_b + E_c)/2$.

The basic idea behind the protocol is that: even if party A and party B conspire to obtain S_{area}, they fail to determine the position of (x_c, y_c) on the line 1 or line 2, because every triangle consisting of each point on the two lines, the point (x_a, y_a) and the point (x_b, y_b), has the equal area (see Figure 1, where both lines are parallel to the line $\overline{AB}$ and have the same distance to it).

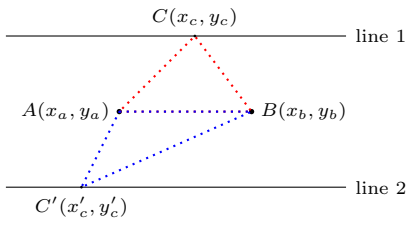


Figure 1: Every point on line 1 or line 2 is a candidate

The scheme takes for granted that every point on line 1 or line 2 is a possible candidate for the party C's input (x_c, y_c), if the lengths E_b, E_a of the segments $\overline{AC}, \overline{BC}$ are not available to the other two parties. In order to protect

Table 1: The sketch of protocol-1

Party	A	B	C
Input	(x_a, y_a)	(x_b, y_b)	(x_c, y_c)
Seeds	$\{r_c, r_{c1}\}, \{r_b, r_{b1}\}$	$\{r_c, r_{c1}\}, \{r_a, r_{a1}\}$	$\{r_a, r_{a1}\}, \{r_b, r_{b1}\}$
Encrypt	$\mathrm{CT}_{acx} = \mathrm{Enc}_{PK_c}(G(r_c)x_a + G(r_{c1}))$ $\mathrm{CT}_{acy} = \mathrm{Enc}_{PK_c}(G(r_c)y_a + G(r_{c1}))$ $\mathrm{CT}_{abx} = \mathrm{Enc}_{PK_b}(G(r_b)x_a + G(r_{b1}))$ $\mathrm{CT}_{aby} = \mathrm{Enc}_{PK_b}(G(r_b)y_a + G(r_{b1}))$	$\mathrm{CT}_{bcx} = \mathrm{Enc}_{PK_c}(G(r_c)x_b + G(r_{c1}))$ $\mathrm{CT}_{bcy} = \mathrm{Enc}_{PK_c}(G(r_c)y_b + G(r_{c1}))$ $\mathrm{CT}_{bax} = \mathrm{Enc}_{PK_a}(G(r_a)x_b + G(r_{a1}))$ $\mathrm{CT}_{bay} = \mathrm{Enc}_{PK_a}(G(r_a)y_b + G(r_{a1}))$	$\mathrm{CT}_{cax} = \mathrm{Enc}_{PK_a}(G(r_a)x_c + G(r_{a1}))$ $\mathrm{CT}_{cay} = \mathrm{Enc}_{PK_a}(G(r_a)y_c + G(r_{a1}))$ $\mathrm{CT}_{cbx} = \mathrm{Enc}_{PK_b}(G(r_b)x_c + G(r_{b1}))$ $\mathrm{CT}_{cby} = \mathrm{Enc}_{PK_b}(G(r_b)y_c + G(r_{b1}))$
Send	$A \xrightarrow{\mathrm{CT}_{abx},\ \mathrm{CT}_{aby}} B$ $A \xrightarrow{\mathrm{CT}_{acx},\ \mathrm{CT}_{acy}} C$	$B \xrightarrow{\mathrm{CT}_{bcx},\ \mathrm{CT}_{bcy}} C$ $B \xrightarrow{\mathrm{CT}_{bax},\ \mathrm{CT}_{bay}} A$	$C \xrightarrow{\mathrm{CT}_{cax},\ \mathrm{CT}_{cay}} A$ $C \xrightarrow{\mathrm{CT}_{cbx},\ \mathrm{CT}_{cby}} B$
Decrypt--derive	$G(r_a)E_a$ $Com_a = \frac{1}{2}G(r_a)E_a \cdot G(r_b) \cdot G(r_c)$	$G(r_b)E_b$ $Com_b = \frac{1}{2}G(r_b)E_b \cdot G(r_a) \cdot G(r_c)$	$G(r_c)E_c$ $Com_c = \frac{1}{2}G(r_c)E_c \cdot G(r_b) \cdot G(r_a)$
Send	see the original step 6 and 7		
Derive	$Mul_a = G(r_a)G(r_b)G(r_c)(P - E_a)$ $G(r_a)G(r_b)G(r_c)P$	$Mul_b = G(r_a)G(r_b)G(r_c)(P - E_b)$ $G(r_a)G(r_b)G(r_c)P$	$Mul_c = G(r_a)G(r_b)G(r_c)(P - E_c)$ $G(r_a)G(r_b)G(r_c)P$
Send	see the original step 9		
Output	$A \xleftarrow{S_{area}} B$	compute S_{area}	$B \xrightarrow{S_{area}} C$

the lengths, it eventually encrypts them as $G(r_a)E_a$ and $G(r_b)E_b$, where r_a is not known to party A, and r_b is not known to party B. The main procedure of this protocol can be described as follows (see Table 1). We refer to [16] for more details.

2.2 It is Insecure Against Collusion Attack

As we know the chief threat to secure multi-party computation is that some participators could form an alliance to try to recover other participators' inputs. Such a conventional collusion attack must be thoroughly eliminated when MPC protocols are designed. But we find the protocol can not resist the conventional collusion attack.

In the security argument of the protocol, it claims that (see §4, page 6, [16]):

> *None of the participants is able to obtain extra information beyond the protocol's provision even when she knows arbitrary number of inputs belonging to other parties.*

But we find the claim is not sound because any two parties can collaborate to recover the other party's input. For example, party A and party B can collaborate to obtain party C's input (x_c, y_c). Actually, in the protocol A knows

$$\{r_c, r_b, G(r_a)E_a, S_{area}\},$$

and B knows

$$\{r_c, r_a, G(r_b)E_b, S_{area}\}.$$

If they form an alliance, they can obtain

$$x_a,\ y_a,\ x_b,\ y_b,\ r_a,\ r_b,\ r_c,\ r_{a1},\ r_{b1},\ r_{c1},\ G(r_a)E_a,\ G(r_b)E_b,\ S_{area}.$$

Since the pseudorandom generator $G(\cdot)$ is public, they can recover E_a, E_b from $G(r_a)E_a, G(r_b)E_b$.

It is clear that the other point (x_c, y_c) lies either on the circle at the center (x_a, y_a) with the radius E_b, or on the circle at the center (x_b, y_b) with the radius E_a. Thus, they only need to solve the equations

$$\begin{cases} (x_a - x_c)^2 + (y_a - y_c)^2 = E_b^2 \\ (x_b - x_c)^2 + (y_b - y_c)^2 = E_a^2 \end{cases}$$

for x_c and y_c. Geometrically, the target point (x_c, y_c) is just one intersection of two circles (Figure 2). Thus, they can definitely recover the point.

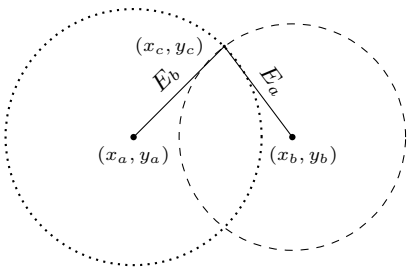


Figure 2: The target point must be one of the intersections of two circles with the radiuses E_a, E_b

3 Analysis of Protocol-2

3.1 Review of Protocol-2

The protocol [16] is based directly on the Shoelace formula, which says that the area of a triangle with vertexes

Table 2: The sketch of protocol-2

Party	A	B	C
Input	(x_a, y_a)	(x_b, y_b)	(x_c, y_c)
Seeds	r_a	r_b	r_c
Encrypt	$\mathrm{CT}_{acx} = \mathrm{Enc}_{PK_c}(G(r_a)x_a)$ $\mathrm{CT}_{acy} = \mathrm{Enc}_{PK_c}(G(r_a)y_a)$ $\mathrm{CT}_{abx} = \mathrm{Enc}_{PK_b}(G(r_a)x_a)$ $\mathrm{CT}_{aby} = \mathrm{Enc}_{PK_b}(G(r_a)y_a)$	$\mathrm{CT}_{bcx} = \mathrm{Enc}_{PK_c}(G(r_b)x_b)$ $\mathrm{CT}_{bcy} = \mathrm{Enc}_{PK_c}(G(r_b)y_b)$ $\mathrm{CT}_{bax} = \mathrm{Enc}_{PK_a}(G(r_b)x_b)$ $\mathrm{CT}_{bay} = \mathrm{Enc}_{PK_a}(G(r_b)y_b)$	$\mathrm{CT}_{cax} = \mathrm{Enc}_{PK_a}(G(r_c)x_c)$ $\mathrm{CT}_{cay} = \mathrm{Enc}_{PK_a}(G(r_c)y_c)$ $\mathrm{CT}_{cbx} = \mathrm{Enc}_{PK_b}(G(r_c)x_c)$ $\mathrm{CT}_{cby} = \mathrm{Enc}_{PK_b}(G(r_c)y_c)$
Send	$A \xrightarrow{\mathrm{CT}_{abx},\ \mathrm{CT}_{aby}} B$ $A \xrightarrow{\mathrm{CT}_{acx},\ \mathrm{CT}_{acy}} C$	$B \xrightarrow{\mathrm{CT}_{bcx},\ \mathrm{CT}_{bcy}} C$ $B \xrightarrow{\mathrm{CT}_{bax},\ \mathrm{CT}_{bay}} A$	$C \xrightarrow{\mathrm{CT}_{cax},\ \mathrm{CT}_{cay}} A$ $C \xrightarrow{\mathrm{CT}_{cbx},\ \mathrm{CT}_{cby}} B$
Decrypt--derive	$G(r_b)x_b, G(r_b)y_b, G(r_c)x_c, G(r_c)y_c$ $G(r_a)G(r_b)G(r_c)(x_by_c - x_cy_b)$	$G(r_a)x_a, G(r_a)y_a, G(r_c)x_c, G(r_c)y_c$ $G(r_a)G(r_b)G(r_c)(x_cy_a - x_ay_c)$	$G(r_b)x_b, G(r_b)y_b, G(r_a)x_a, G(r_a)y_a$ $G(r_a)G(r_b)G(r_c)(x_ay_b - x_by_a)$
Send	see the original step 5 and 6		
Output	$A \xleftarrow{S_{area}} C$	$B \xleftarrow{S_{area}} C$	compute S_{area}

$(x_a, y_a), (x_b, y_b), (x_c, y_c)$ is

$$S_{area} = \frac{1}{2}(x_ay_b - x_by_a + x_by_c - x_cy_b + x_cy_a - x_ay_c).$$

Its main steps can be described as follows (see Table 2).

3.2 It is Not Immune to Collusion Attack

The protocol is insecure against collusion attack. For example, party A and party B can collaborate to recover (x_c, y_c). Notice that in the protocol A knows

$$r_a, G(r_b)x_b, G(r_b)y_b, G(r_c)x_c, G(r_c)y_c, S_{area},$$

and B knows

$$r_b, G(r_a)x_a, G(r_a)y_a, G(r_c)x_c, G(r_c)y_c, S_{area}.$$

If they form an alliance, they can obtain

$$x_a,\ y_a,\ x_b,\ y_b,\ r_a,\ r_b,$$
$$X_c := G(r_c)x_c,\ Y_c := G(r_c)y_c,\ S_{area}.$$

Hence, they only need to solve the following equations

$$\begin{cases} S_{area} = \frac{1}{2}(x_ay_b - x_by_a + x_by_c - x_cy_b + x_cy_a - x_ay_c) \\ X_c = G(r_c)x_c \\ Y_c = G(r_c)y_c \end{cases}$$

for $G(r_c), x_c$, and y_c. Therefore,

$$\begin{aligned} G(r_c) &= \frac{x_bY_c - X_cy_b + X_cy_a - x_aY_c}{2S_{area} - x_ay_b + x_by_a} \\ x_c &= \frac{X_c}{G(r_c)} \\ y_c &= \frac{Y_c}{G(r_c)} \end{aligned}$$

Although party A and party B cannot obtain the random seed r_c which is chosen by party C, they can successfully recover the point (x_c, y_c). The original designers had forgotten to check the possibility of solving the above equations, and simply claimed that the protocol-2 was secure against collusion attack.

4 Conclusion

We show that the two secure three-party computation protocols for triangle area are not sound because they cannot resist the conventional collusion attack, which is the chief threat to any MPC protocol. We would like to stress that collusion attack should be considered carefully in designing MPC protocols. Besides, some sophisticated cryptographic tools should be integrated tactfully. To the best of our knowledge, the special instance (secure three-party computation for triangle area) of secure multi-party computation remains unsolved.

Acknowledgements

We thank the National Natural Science Foundation of China (Project 61411146001). We are grateful to the reviewers for their valuable suggestions.

References

[1] I. Blake and V. Kolesnikov, "Strong conditional oblivious transfer and computing on intervals," in *Proceedings of 10th International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology*, pp. 515–529, Dec. 2004.

[2] Z. Brakerskim and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[3] Z. J. Cao and H. Y. Cao, "Improvement of camenisch-neven-shelat oblivious transfer scheme," *International Journal of Network Security*, vol. 17, no. 2, pp. 103–109, 2015.

[4] 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.
[5] Z. J. Cao, L. H. Liu, and O. Markowitch, "Comment on 'highly efficient linear regression outsourcing to a cloud'," *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 893, 2019.
[6] G. Castagnos and F. Laguillaumie, "Linearly homomorphic encryption from DDH," in *Proceedings of The Cryptographer's Track at the RSA Conference*, pp. 487–505, Apr. 2015.
[7] J. H. Cheon and J. Kim, "A hybrid scheme of public-key encryption and somewhat homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1052–1063, 2015.
[8] S. F. Chiou, H. T. Pan, E. F. Cahyadi, and M. S. Hwang, "Cryptanalysis of the mutual authentication and key agreement protocol with smart cards for wireless communications," *International Journal of Network Security*, vol. 21, no. 1, pp. 100–104, 2019.
[9] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM*, vol. 28, no. 6, pp. 637–647, 1985.
[10] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09)*, pp. 169–178, June 2009.
[11] M. Green and S. Hohenberger, "Practical adaptive oblivious transfer from simple assumptions," in *Proceedings of 8th Theory of Cryptography Conference (TCC'11)*, pp. 347–363, Mar. 2011.
[12] L. C. Huang, T. Y. Chang, and M. S. Hwang, "A conference key scheme based on the Diffie-Hellman key exchange," *International Journal of Network Security*, vol. 20, no. 6, pp. 1221–1226, 2018.
[13] K. Kurosawa and R. Nojima, "Simple adaptive oblivious transfer without random oracle," in *Proceedings of 15th International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology*, pp. 334–346, Dec. 2009.
[14] H. Lin and W. G. Tzeng, "An efficient solution to the millionaires'problem based on homomorphic encryption," in *Proceedings of 3rd International Conference on Applied Cryptography and Network Security (ACNS'05)*, pp. 456–466, June 2005.
[15] T. C. Lin, T. Y. Yeh, and M. S. Hwang, "Cryptanalysis of an ID-based deniable threshold ring authentication," *International Journal of Network Security*, vol. 21, no. 2, pp. 298–302, 2019.
[16] L. Liu, X. F. Chen, and W. J. Lou, "Secure three-party computational protocols for triangle area," *International Journal of Information Security*, no. 15, pp. 1–13, 2016.
[17] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation," in *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC'99)*, pp. 245–254, May 1999.
[18] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Proceedings of 19th Annual International Cryptology Conference, Advances in Cryptology*, pp. 573–590, Aug. 1999.
[19] M. Naor and B. Pinkas, "Computationally secure oblivious transfer," *Journal of Cryptology*, vol. 18, no. 1, pp. 1–35, 2005.
[20] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceeding of International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology*, pp. 223–238, May 1999.
[21] H. T. Pan, E. F. Cahyadi, S. F. Chiou, and M. S. Hwang, "Research on batch verification schemes for identifying illegal signatures," *International Journal of Network Security*, vol. 21, no. 6, pp. 1062–1070, 2019.
[22] B. Schoenmakers and P. Tuyls, "Pratical two-party computation based on the conditional gate," in *Proceedings of 10th International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology*, pp. 119–136, Dec. 2004.
[23] W. G. Tzeng, "Efficient 1-out-of-n oblivious transfer protocols with universally usable parameter," *IEEE Transactions on Computers*, vol. 53, no. 2, pp. 232–240, 2004.
[24] A. C. C. Yao, "Protocols for secure computations," in *Proceedings of 23rd Annual Symposium on Foundations of Computer Science (FOCS'82)*, pp. 160–164, Nov. 1982.
[25] A. C. C. Yao, "How to generate and exchange secrets," in *Proceedings of 27th Annual Symposium on Foundations of Computer Science (FOCS'86)*, pp. 162–167, Oct. 1986.
[26] B. Zeng, C. Tartary, P. Xu, and *et al.*, "A practical framework for t-out-of-n oblivious transfer with security against covert adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 465–479, 2012.

Lihua Liu, associate professor with Department of Mathematics at Shanghai Maritime University, received her PhD degree in applied mathematics from Shanghai Jiao Tong University. Her research interests include combinatorics and cryptography.

Jie Cao is currently pursuing his bachelor degree from School of Computer Science and Technology, Soochow University. His research interests include information security and cryptography.

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.