IJNS
International Journal of Network Security

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

Vol. 22, No. 6 (November 2020)

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: 22, No: 6 (November 1, 2020)

International Journal of Network Security

Research on Blockchain Technologies in Bidding Systems

Yi-Hui Chen[1,2], Li-Chin Huang[3], Iuon-Chang Lin[4], and Min-Shiang Hwang[5,6]
(Corresponding author: Min-Shiang Hwang)

Department of Information Management, Chang Gung University, Taoyuan 33302, Taiwan [1]
Kawasaki Disease Center, Kaohsiung Chang Gung Memorial Hospital, Kaohsiung 83301, Taiwan[2]
(Email: cyh@gap.cgu.edu.tw)
Department of Information Management, Executive Yuan, Taipei 10058, Taiwan[3]
Department of Management Information Systems, National Chung Hsing University, Taiwan[4]
Department of Computer Science & Information Engineering, Asia University, Taichung, Taiwan[5]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taichung, Taiwan, R.O.C.
Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan[6]
(Email: mshwang@asia.edu.tw)
(Received Apr. 13, 2020; Revised and Accepted June 21, 2020; First Online June 30, 2020)

Abstract

Due to the popularity of the Internet, people are increasingly accepting the integration of electronic service applications. Whether it is communication, trading, or transportation, these have gradually changed people's lifestyles. Electronic auctions have also become one of the popular e-commerce activities. Electronic auction systems usually include bidders, auctioneers, and third parties that allow bidders to bid via the Internet. It replaces the inconvenience and low efficiency of traditional tendering. Electronic auctions can be divided into two types: open bidding and sealed bidding. The public bidding method is to continuously increase the bidding price until no bidder is willing to pay a higher bid. The deadline has arrived. The highest bidder is the winner of the public tender. Since bidders can bid multiple times, this bidding method is also called multiple bidding. The bidding method for sealed bids is that the bidder can only send the bill once. Once the deadline arrives, the auctioneer will compare all bills. The bidder with the highest bid is the winner of the "sealed bid". Since bidders can only bid once, this bidding method is also called a single bid auction. Both bidding methods have their practicability. But no matter what kind of bidding. It should rely on intermediaries to allow buyers and sellers to conduct transactions. Lead to trust and transaction cost issues. In this regard, we will use blockchain technology to develop smart contracts for public bidding and sealed bidding. It uses the characteristics of blockchain decentralization and low transaction costs to improve the shortcomings of electronic auctions.

Keywords: Bid; Blockchain; E-auction; P2P Network; Public Bid; Sealed Smart Contract

1 Introduction

Due to the popularity of the Internet, most people have gradually accepted electronic integrated applications. Whether it is communication, transaction, or service, it has profoundly changed people's living habits. Electronic bidding has become one of the popular e-commerce activities [1, 8].

Electronic auctions originated from traditional auctions. It is an application that combines Internet technology and auction mechanism to speed up transaction efficiency and speed [2, 3, 14, 18, 23]. It is a trading system that breaks the limitations of time, space, and geography through Internet technology. Therefore, electronic bidding has become an incredibly popular transaction mode in e-commerce.

Electronic bidding is usually composed of bidders, auctioneers, and third parties (see Figure 1). Currently, most e-bidding systems are mainly provided by intermediaries to provide platforms and services. Buyers and sellers can publish, bid, or trade. Popular auction platforms include Yahoo auctions, open-air auctions, and shrimp auctions. However, due to the current need to rely on intermediaries' platforms and services, intermediaries must pay some fees, such as publishing fees, transaction fees, etc. It may cause the problem of increased transaction costs between the buyer and seller [19, 20, 22]. Therefore, this research applies blockchain smart contract technology to electronic bidding [4–7, 15–17, 21, 25]. The use of the blockchain's decentralized nature eliminates the intermediary in electronic bidding, so buyers and sellers can

directly conduct transactions without relying on intermediaries.

Figure 1: The entities in the electronic bidding system

2 Types of Bidding

Electronic bidding can be divided into three main types of bidding:

1) English Auction:
English auctions, also known as price-raising auctions, are the most common and frequently encountered auction method [9,12,24]. During the auction, the price suggested by all bidders must be higher than the previous price. When the auction time expires, the highest bidder will get the item. However, "Sniping" often occurs in online auctions. In other words, until the last few minutes before the auction ends, a specific bidder makes a bid. So there is no time for the remaining bidders to fight back. The solution to this phenomenon is to add an "expansion period" before the original fixed period. For example, if the extension time is set to ten minutes, it means that in the last ten minutes, if there are any bidders, the auction deadline will be automatically extended by ten minutes. This method effectively solves the sniper phenomenon.

2) Dutch Auction:
Dutch auctions are also called reduced price auctions. After a specific time interval, the main feature is that the price will be reduced according to the initially set price reduction rules until the bidder is willing to buy at that price [10,11,13]. This action is more suitable for perishable items such as fruits and vegetables.

   In English auctions, the initial price of the product is usually lower than its market price. After bidders bid with each other, the price will be close to the market price. As prices increase, the number of bidders will also decrease. Dutch auctions are the opposite of English auctions. The initial price of the commodity will be higher than its market price. As the price drops, the number of bidders will increase.

3) Sealed Bid Auction:
In the sealed bid list, the prices of all bidders will be sealed. The prices of all bidders will not be compared before the deadline for the bid opening [8, 14, 18]. Electronically sealed bidding auctions often have a common flaw. Before the bid opening deadline, bidders cannot ensure that their bid prices have been leaked by third parties (leading bidders), which may result in malicious bidders colluding with leading bidders to obtain the best bid price. The research topic aims to use blockchain smart contract technology to ensure the confidentiality, non-repudiation, and non-changeability of electronically sealed bids and solve electronically sealed bids' shortcomings.

With the current development of electronic bidding, two main problems can be found. First of all, the transaction process of electronic bidding must rely on intermediary agencies. It is difficult for buyers and sellers to communicate directly. It also causes problems such as increased transaction costs. Therefore, this research proposes three research topics:

1) Applying blockchain technology to electronic bidding. Using the blockchain's decentralized nature, intermediaries that were originally indispensable for e-bidding have been deleted to reduce transaction costs.

2) In sealed bidding, the bidder cannot ensure that the lousy bidder leaks the bid price. The protection of fair competition may be less. Therefore, this research topic aims to use blockchain smart contracts to improve the shortcomings of sealed bidding. Use the blockchain's immutability to write rules in a sealed bid so that no one can open it before the bid opening time comes to ensure the data's privacy.

3) Use private blockchain to conduct related research on public bidding and sealed bidding.

3 Research on Blockchain Bidding Systems

Due to the rapid development of Internet technology, electronic bidding has replaced traditional bidding. The inconvenient and ineffective bidding mechanism of traditional bidding has been improved. It allows bidders to bid through the Internet anytime and anywhere. Provide bidders and bidders with a faster and more convenient transaction mechanism. However, in the current electronic bidding transaction mechanism, it is necessary to rely on intermediaries to complete the two parties' transactions. Therefore, the following two problems may occur:

1) Trust Issues:
To complete a transaction through an intermediary, you may first need to use personal data to apply for a set of accounts that can be used to use the platform's services. After the transaction is completed, the account's transaction details will be stored in the platform's database. Users may worry that their personal information or transaction records will be leaked out, causing mistrust.

2) Transaction Cost Issue:
In the transaction process, to use the platform's ser-

vices, users may need to pay some platform publishing fees, advertising fees, or transaction fees. It may lead to higher transaction costs between buyers and sellers.

In response to the above two electronic bidding issues, this research aims to use the characteristics of blockchain decentralization and zero trust foundation to develop an electronic bidding system based on blockchain smart contract applications to solve the trust problem and reduce transaction costs. The following subsections will illustrate the implementation methods and steps of these research topics.

3.1 Research on Smart Contracts for Public Bidding

A complete public bidding e-bidding mechanism has the following basic requirements:

1) The identity of the bidder during the bidding process is anonymous. After the bidding is over, the bidders and successful bidders are anonymous.
2) During the sending process, the content of the bid list cannot be changed. Everyone can verify the source of the bid and the correctness and completeness of the content.
3) No one can pretend to be a legitimate bidder. After bidding, the bidder cannot deny that the bid has been submitted.
4) The bidder must prove that he has submitted his bid or prove that he has won the bid.
5) After winning the bid, the bidder can ask for money from the winning bidder, but the bidder cannot ask for money from the winning bidder.

The electronic bidding process of public bidding is shown in Figure 2. In the beginning, the bidding meeting announced bidding information, including product descriptions and starting prices. After that, the bidder can continue to bid. The bidder will receive the bid submitted by the bidder. And send a message to the bidder notifying that the bid has been received. And announce the current highest bid to everyone. Before any bidder offers a higher price, the bidder will announce the final bid price. And collect money from the winning bidder, and send the goods to the winning bidder after confirmation.

The first research topic is the study of smart contracts for public bidding on the blockchain. This research will develop public bidding through blockchain smart contract development. Write the public bidding transaction contract on the blockchain. Use peer-to-peer technology to achieve the purpose of decentralization. All bidders can bid by calling this public bidding transaction contract without relying on intermediary agencies. To this end, the steps of this study are as follows:

1) Create an Account:
   The process of creating an account. Use the Ethereum wallet to create two blockchain accounts to facilitate subsequent testing, transactions, etc.
2) Mining:
   Use the command line and MinerGate to perform mining. Get currency to pay commissions when creating contracts and transactions.
3) Perform Block Synchronization and View Block Status:
   At this stage, use the command line to synchronize the blocks. After that, you can enter a block to view the detailed information in that block.
4) Create a Smart Contract:
   Establishing a smart contract is mainly divided into three execution steps: writing, compiling and deploying. Use Sublime to write the Solidity programming language. Then use the Solidity real-time compiler and runtime to compile the agreement into Bytecode and Interface. Finally, use the Ethereum wallet to deploy the contract and publish the contract to the blockchain.
5) Test Contract:
   When the contract is verified in the contract testing phase, the contract's address can be obtained. Use the previously created Account 2 wallet and the interface obtained by compiling Solidity to add the contract to test its related functions.
6) The Structure of Smart Contracts for Public Bidding:
   Figure 3 shows the program structure of the open bid smart contract in our system. The contract is mainly divided into initialization and contract functions.

In the initialization data, the following data will be announced in advance:

- Auctioneer: It is used to record the address of the bidder who initiated the contract.
- AuctionStart: It is used to announce the start time of the bidding.
- bidding time: It is used to announce the significant time of the contract.
- maximumBidder: It is used to record the address of the current highest bidder.
- maximumBid: It is used to record the current maximum price.

In the contract function, the following information will be announced in advance:

- Bid(): Anyone can call this function to perform bidding operations. Before executing this function, first, use AuctionStart and bid time to determine whether the contract expires. If not, the bidder can enter the

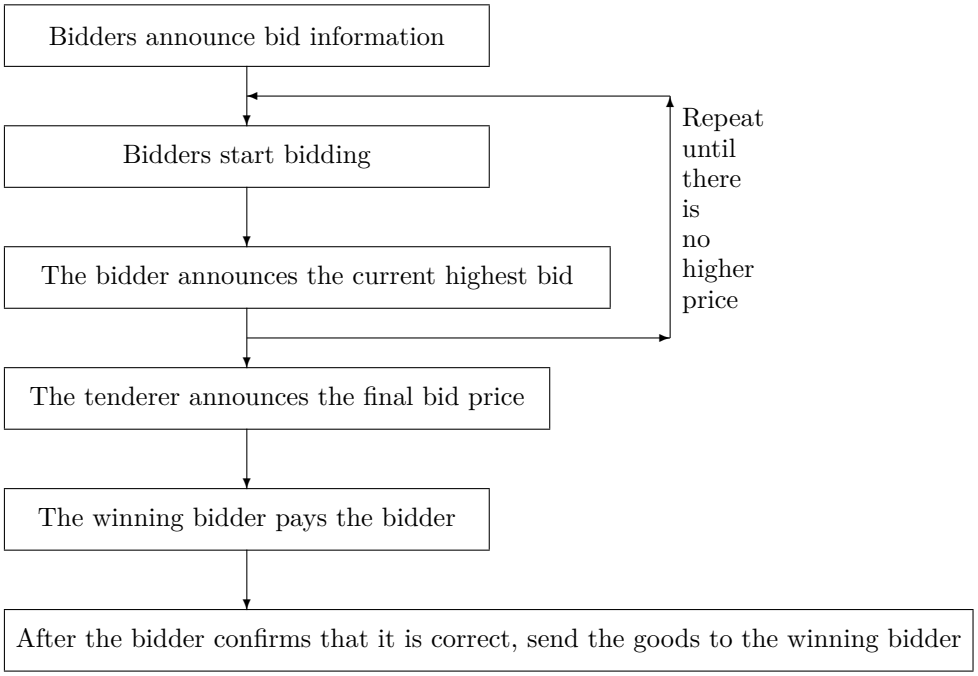


Figure 2: Public bidding process

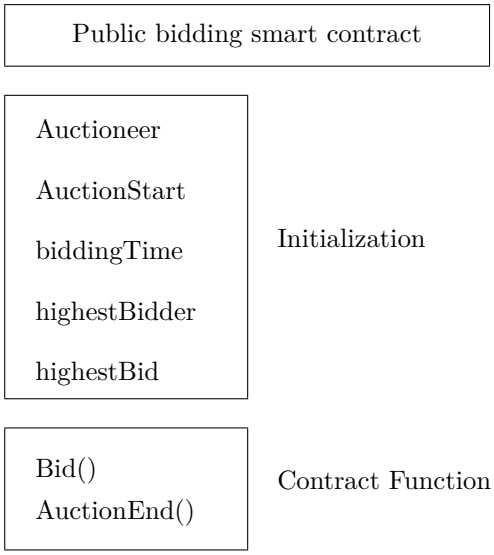


Figure 3: Contract structure of public bidding

bid amount. If the price is greater than the current highest price, the highest bid and highest bidder will be used to record the bidder's amount and address.

- AuctionEnd(): In this function, AuctionStart and bid time are automatically used to determine the contract's sufficient time. If the significant time is over, the winning bidder's address and the amount will be automatically sent to the bidder. This function will be turned off to avoid repeated execution.

It is expected that potential problems will be encountered when conducting this research. The following is an explanation of the solutions to these problems: In a public tender, each bidder can make multiple bids. Therefore, during contract testing, multiple accounts need to be used for mutual bidding. On the public chain, a small fee is required for each execution of an instruction. Therefore, mining operations must be performed on each account. In this way, too much time is spent testing the contract. To this end, this research will first use a testnet to test the contract. Save mining time for each account. After the test is completed, deploy the completed smart contract on the leading blockchain network.

3.2 Research on Smart Contracts with Sealed Bids

A complete sealed bidding mechanism has the following basic requirements:

1) Throughout the bidding process, the identities of bidders and successful bidders are anonymous.

2) During the transfer process, the content of the bid list cannot be changed. Everyone can verify the source of the bid and the correctness and completeness of the content.

3) No one can pretend to be a legitimate bidder. After bidding, the bidder cannot deny that the bid has been submitted.

4) The bidder must prove that he has submitted his bid or prove that he has won the bid.

5) Bids must be delivered within a significant time; expired bids are invalid.

6) Before the bid is opened, no one can open the bid.

7) When encountering the same price, there must be a fair solution.

The bidding process for sealed bidding is shown in Figure 4. In the beginning, the bidders announced the bid information, including product descriptions and starting prices. Bidders who want to participate in the bidding must first register with the exhibition agency. After the identity is confirmed, all legal bidders can bid before the deadline for bidding. All bids will be encrypted and then sent to bidders. All encrypted bids will be unlocked before the bid opening, and the final winner can be determined after comparison.

In the sealed bidding mechanism, it is necessary to rely on impartial intermediary agencies to assist. However, it is also possible that the tender and the impartial agency may conspire to disclose bid information in the bid before the bid opening. Therefore, the second research topic is the study of smart contracts with sealed bidding on the blockchain. Sealed bidding will be developed through blockchain smart contracts. Use the decentralized function of the blockchain to improve this problem. The steps of this research are as follows:

1) Extend the first research topic into mining, contract writing, contract preparation, and contract deployment.

2) Seal the structure of the smart bidding contract. Figure 5 shows the program structure of the sealed bid smart contract in the research system. The contract is mainly divided into initialization and contract functions.

In the initialization data, we declare the following data in advance:

- auctioneer:It is used to record the address of the bidder who initiated the contract.
- auctionStart: It is used to announce the start time of the auction.
- biddingTime: It is used to announce the contract time.
- bididngEnd: It is used to announce the bidding time of the contract.
- maximumBidder: It is used to record the address of the current highest bidder.
- maximumBid: It is used to record the current maximum price.

In the contract function, the following information will be announced in advance:

- blindAuction(): Activate the contract by calling this function. And use auctionStart and biddingEnd to record the start and end time.
- bid(): The bidder can call this function to perform bidding operations.
- Reveal (): Perform bid opening action by calling this function. And compare the prices of all bids to get the final bidder.
- auctionEnd(): By calling this function, the number and addresses of successful bidders will be collected.
- withdraw(): Refund the bid amount of someone other than the winning bidder.

It is expected that potential problems will be encountered in the implementation of this research. The following is an explanation of these problems: In a sealed bid smart contract, the contract's function is more complicated. For bidders and bidders, the wrong contract function may be called. For example, the bidder accidentally called show() to open all bids, so bids must be terminated and redeployed. To this end, this research will establish authority judgments for different functions. Before executing this function, it will determine whether the caller can execute this function.

3.3 Research on Public Bidding and Private Chain Public Bidding

The public chain is an open node. Anyone can enter other nodes to view transaction status and contract content. Therefore, the privacy of smart contract code may not be protected and may be abused by others. Therefore, the study of public bidding and sealed bidding on the private chain will use the private chain to develop smart contracts for public bidding and sealed bidding. Use the command line to create a dedicated chain. Use the characteristics of a dedicated chain to control the write and read permissions of the node. They are used to protect private transaction records and smart contract content.

It is expected that this study will encounter some potential problems. The following is an explanation of the solutions to these problems: At present, most blockchain

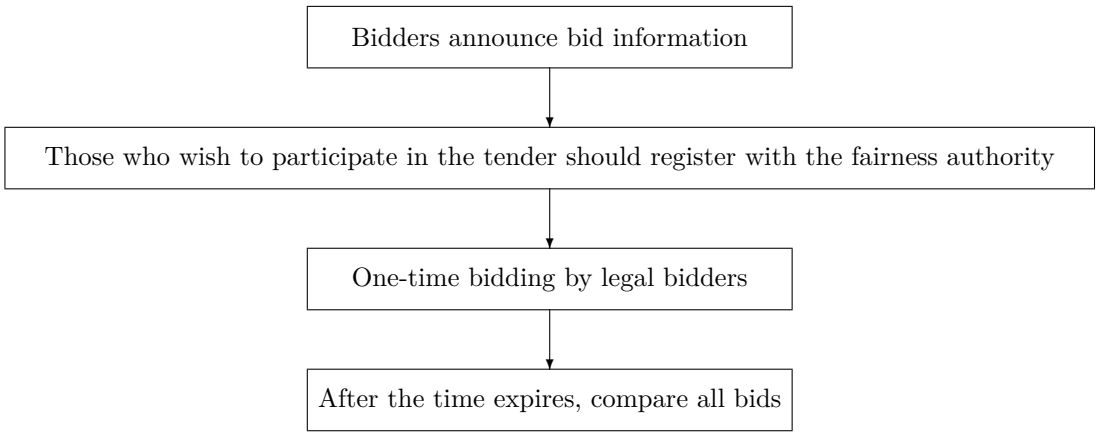


Figure 4: The bidding process for a sealed bid

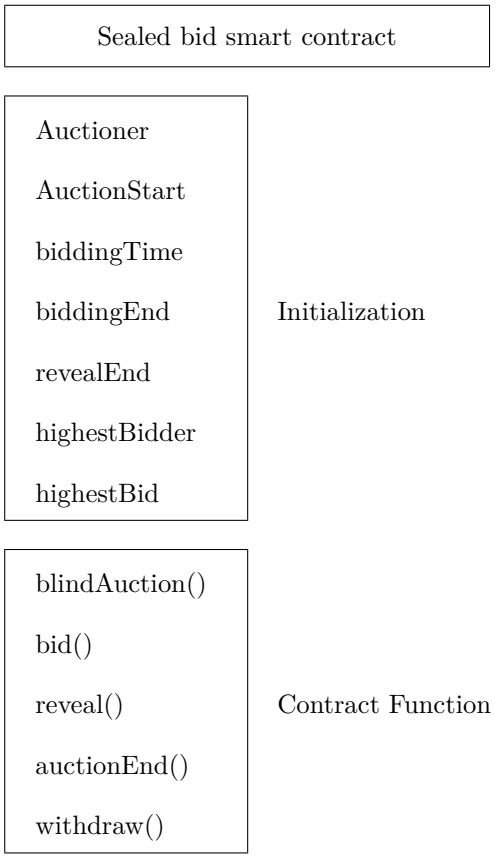


Figure 5: Contract structure of a sealed bid

smart contracts focus on the research of public chains, and there is less literature on private chain smart contracts. Therefore, there is less content to be cited. It requires repeated research to find the best solution. This research aims at the application of Ethereum token in private chain contract testing.

4 Conclusions

In this article, we have proposed three research topics: 1) Research on smart contracts for public bidding; 2) Research on smart contracts with sealed bids; 3) Research on public bidding and private chain public bidding.

This research deployed smart contracts on public and private chains. Use the respective characteristics and advantages of public and private chains to develop applications suitable for different fields. This research uses blockchain smart contracts to improve the shortcomings of the e-bidding system. Replace the transaction mechanism that previously relied on a third party. It not only solves the issue of trust enforcement between the parties to the transaction. It can also reduce transaction costs. The results of this research can be applied in many fields. For example, it can be applied to cross-border remittances, contract insurance policies, and loan credit in the financial market. The industrial supply chain can be applied to product history tracking and supply a collaborative chain supply.

Acknowledgments

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

References

[1] G. Cao and J. Chen, "Practical electronic auction scheme based on untrusted third-party," in *International Conference on Computational and Information Sciences*, pp. 493-496, 2013.

[2] T. S. Chandrashekar, Y. Narahari, C. H. Rosa, D. M. Kulkarni, J. D. Tew and P. Dayama, "Auction-based mechanisms for electronic procurement," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 3, pp. 297-321, 2007.

[3] W. Chen and F. Lei, "A simple efficient electronic auction scheme," in *Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'07)*, pp. 173-174, 2007.

[4] Y. H. Chen, L. C. Huang, I. C. Lin, and M. S. Hwang, "Research on the secure financial surveillance blockchain systems," *International Journal of Network Security*, vol. 22, no. 4, pp. 708-716, 2020.

[5] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292-2303, 2016.

[6] 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.

[7] C. Hu, D. Zheng, R. Guo, A. Wu, L. Wang, and S. Y. Gao, "A novel blockchain-based anonymous handover authentication scheme in mobile networks," *International Journal of Network Security*, vol. 22, no. 5, pp. 874-884, 2020.

[8] X. Hu, Z. Qin, F. Zhang, Y. Yang, and Y. Zhao, "A sealed-bid electronic auction protocol based on ring signature," in *International Conference on Communications, Circuits and Systems*, pp. 480-483, 2007.

[9] M. S. Hwang, E. J. L. Lu, I. C. Lin, "Adding timestamps to the secure electronic auction protocol", *Data & Knowledge Engineering*, vol. 40, no. 2, pp. 155-162, Feb. 2002.

[10] W. Jiang, J. Chen, Y. Xu, Y. Wang, L. Tan, "Research on the influence factors of consumer repurchase in Dutch auction", *Lecture Notes in Computer Science*, vol. 11354, pp. 330-338, 2019.

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

[12] C. C. Lee, M. S. Hwang, C. W. Lin, "A new English auction scheme using the bulletin board system", *Information Management and Computer Security*, vol. 17, no. 5, pp. 408-417, Nov. 2009.

[13] C. C. Lee, M. S. Hwang, C. W. Lin, "An efficient multi-round anonymous auction protocol", *Journal of Discrete Mathematical Sciences & Cryptography*, vol. 10, no. 4, pp. 547-557, Aug. 2007.

[14] S. Li, X. Li, M. X. He, S. K. Zeng and X. L. Tang, "Sealed-BID electronic auction without the third party," in *11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP'14)*, pp. 336-339, 2014.

[15] Z. C. Li, J. H. Huang, D. Q. Gao, Y. H. Jiang and F. Li, "ISCP: An improved blockchain consensus protocol," *International Journal of Network Security*, vol. 21, no. 3, pp. 359-367, 2019.

[16] 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.

[17] Y. Liu, M. He, and F. Pu, "Anonymous transaction of digital currency based on blockchain," *International Journal of Network Security*, vol. 22, no. 3, pp. 444-450, 2020.

[18] W. Shi, I. Jang and H. S. Yoo, "A sealed-bid electronic marketplace bidding auction protocol by using ring signature," in *Fourth International Conference on Computer Sciences and Convergence Information Technology*, pp. 1005-1009, 2009.

[19] W. K. Tan and Y. L. Chung, "User payment choice behavior in e-auction transactions," in *International Conference on e-Education, e-Business, e-Management and e-Learning*, pp. 183-187, 2010.

[20] C. C. Tu, C. Y. Lin and K. Fang, "Customers' perception on attitude towards e-auction," in *IEEE Asia-Pacific Services Computing Conference*, pp. 1044-1048, 2008.

[21] L. Wang, D. Zheng, R. Guo, C. Hu, and C. M. Jing, "A blockchain-based privacy-preserving authentication scheme with anonymous identity in vehicular networks," *International Journal of Network Security*, vol. 22, no. 6, pp. 981-990, 2020.

[22] S. Yao, W. A. Cui and Z. Wang, "A model in support of bid evaluation in multi-attribute e-auction for procurement," in *4th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1-4, 2008.

[23] F. Zhang, Q. Li and Y. Wang, "A new secure electronic auction scheme," in *IEEE/AFCEA Information Systems for Enhanced Public Safety and Security (EUROCOMM'00)*, pp. 54-56, 2000.

[24] H. Zhong, S. Li, T. F. Cheng, C. C. Chang, "An efficient electronic English auction system with a secure on-shelf mechanism and privacy preserving", *Journal of Electrical and Computer Engineering*, vol. 2016, 2016.

[25] Y. Zhu, R. Guo, G. Gan and W. T. Tsai, "Interactive incontestable signature for transactions confirmation in bitcoin blockchain," in *IEEE 40th Annual Computer Software and Applications Conference (COMPSAC'16)*, pp. 443-448, 2016.

Biography

Yi-Hui Chen received her Ph.D. degree in computer science and information engineering at the National Chung Cheng University. Later on, she worked at Academia

Sinica as a post-doctoral fellow. Then, she worked at IBM's Taiwan Collaboratory Research Center as a Research Scientist, the Department of M-Commerce and Multimedia Applications, Asia University as an associate professor. She is now an associate professor at the Department of Information Management, Chang Gung University. Her research interests include multimedia security, semantic web, text mining, and multimedia security.

Li-Chin Huang received the B.S. in computer science from Providence University, Taiwan, in 1993 and M.S. in information management from Chaoyang University of Technology (CYUT), Taichung, Taiwan, in 2001; and the Ph.D. degree in computer and information science from National Chung Hsing University (NCHU), Taiwan in 2001. Her current research interests include information security, cryptography, medical image, data hiding, network, security, big data, and mobile communications.

Iuon-Chang Lin received the B.S. in Computer and Information Sciences from Tung Hai University, Taichung, Taiwan, Republic of China, in 1998; the M.S. in Information Management from Chaoyang University of Technology, Taiwan, in 2000. He received his Ph.D. in Computer Science and Information Engineering in March 2004 from National Chung Cheng University, Chiayi, Taiwan. He is currently a professor of the Department of Management Information Systems, National Chung Hsing University, and Department of Photonics and Communication Engineering, Asia University, Taichung, Taiwan, ROC. His current research interests include electronic commerce, information security, cryptography, and mobile communications.

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 professor and Chairman of the Department of Management Information Systems, NCHU, during 2003-2009. He was also a visiting professor at the University of California (UC), Riverside, and UC. Davis (USA) during 2009-2010. He was a distinguished professor of the Department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Excellent Research Award of National Science Council (Taiwan). Dr. Hwang was a dean of 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 and data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.

A Modified Homomorphic Encryption Method for Multiple Keywords Retrieval

Xiaowei Wang[1], Shoulin Yin[1], Hang Li[1], Lin Teng[1], and Shahid Karim[2]
(Corresponding author: Shoulin Yin)
Software College, Shenyang Normal University[1]
Shenyang 110034, China
Department of Computer Science, ILMA University[2]
Karachi, Pakistan
(Email: 352720214@qq.com)
(Received Apr. 29, 2019; Revised and Accepted Nov. 5, 2019; First Online Apr. 19, 2020)

Abstract

Currently, many data owners choose to outsource the data storage to cloud service providers. Data owners store the local private data with plaintext form in the cloud server. It is difficult to guarantee the data privacy and security. So data owners usually encrypt the local private data and upload it into cloud server. The traditional multi-keyword ciphertext retrieval methods cannot take both accuracy and security into consideration. Therefore, we propose a modified homomorphic encryption method for multiple keywords retrieval in this paper. It can effectively solve the privacy leakage of search keywords problem. The retrieval performance is greatly improved in multi-keyword retrieval process. Experimental results show that this new scheme is more efficient and accurate than other ciphertext retrieval schemes.

Keywords: Homomorphic Encryption; Multi-keyword Ciphertext Retrieval; Private Data

1 Introduction

Cloud storage [16] is a new network storage technology which is the extension and development of cloud computing [3]. It uses the grid technology, distributed file system, cluster application, virtualization functions and software to control network in large-scale heterogeneous storage devices and provide on-demand services to remote users of mass data storage access and processing functions. Cloud storage greatly saves the cost of storage hardware and software, reduces the investment of maintenance personnel, and has outstanding advantages such as low cost and high utilization rate. At the same time, professional cloud storage service providers have unparalleled technology and management level [8, 11], which can provide users with better data security services such as redundant backup and disaster recovery. In order to prevent the disclosure of privacy, users usually encrypt the data and then upload them into the cloud storage platform, and keep the decryption key by themselves such as government documents, national medical and health data, business secrets related documents, personal privacy data, etc [7, 12]. However, the encrypted data will lose some characteristics of the original plaintext such as order and similarity. As a result, the traditional plaintext based retrieval schemes cannot work well in the encryption cloud storage system. For this reason, researchers propose a secure ciphertext based on retrieval problem. The method is divided into two classes:

1) The indexing file-based retrieval method [13, 18];

2) The matching-based retrieval method [14, 17]. However, the two methods either need to maintain complex index structure or have low retrieval efficiency, which is difficult to meet the retrieval requirements of massive ciphertext data in the cloud storage environment. The encrypted data also brings new problems. For example, how can data user retrieve the data of interest quickly when facing massive ciphertext data. A common solution is to use keywords for retrieval.

Single-keyword retrieval cannot meet the needs of users for accurate retrieval, so the research of multi-keyword retrieval emerges at the right moment. Lu [9] proposed a secure sorting keyword retrieval algorithm, scoring documents containing keywords, and using one-to-many order-preserving mapping algorithm to encrypt data. The scheme improved the retrieval efficiency, but reduced the security of data and the precision of retrieval. Li [4] proposed a multi-keyword based on Boolean query scheme, which generated search results based on whether or not the keyword was included. However, this scheme could not distinguish the degree of relevance of multiple documents containing the same keyword. Hu [2] proposed a multi-keyword searchable public key encryption scheme,

but it was later proved that it could not resist keyword guessing attacks. Teng [6] proposed a secure ordering multi-keywords retrieval algorithm, and adopted the coordinate calculation keyword matching scheme and the correlation between document, using the inner product similarity evaluation of correlation between keywords, but the solution method of Boolean made documents that contained the same number of keywords score the same, with no guarantee of accuracy. Therefore, we propose an modified homomorphic encryption method for multiple keywords retrieval in this paper. It can effectively solve the privacy leakage of search keywords problem. The retrieval performance is greatly improved in multi-keyword retrieval.

2 Preliminaries

2.1 Homomorphic Encryption

Homomorphic encryption [5, 19] is an encryption method that can directly process ciphertext data. Under the premise of effectively protecting the privacy of user sensitive data, the homomorphic operations such as addition and multiplication can be directly implemented on the ciphertext, and maintain the plaintext order of the ciphertext when operating. However, homomorphic encryption scheme requires a great deal of computational overhead. How to design a homomorphic ciphertext retrieval scheme with less computational overhead and convenient for indexing is a difficult point in current research.

Using homomorphic encryption technology, it can guarantee the ciphertext algebraic operation results and the same in plaintext encrypted algebraic operation. That is, for any valid operation f and plaintext m, there is $f(Enc(m)) = Enc(f(m))$. This special property allows third parties to perform algebraic operations on ciphertext. No decryption is required. Its significance lies in fundamentally solving the confidentiality problem of data and its operation is entrusted to a third party. At present, there are three main frameworks for constructing full homomorphic encryption.

1) GCD problem. First, a partial homomorphic encryption scheme is constructed. And then the decryption circuit is compressed to perform homomorphic decryption of the ciphertext. So it can achieve the aim of controlling ciphertext noise growth. Finally, a fully homomorphic encryption scheme is obtained under the assumption of cyclic security.

2) R-LWE problem. First, a partial homomorphic encryption scheme is constructed [14]. After ciphertext calculation, key exchange is used to control the expansion of ciphertext vector dimension, and mode exchange is used to control the increase of noise. Without using homomorphic decryption technology, a layered all-homomorphic encryption scheme can be obtained.

3) LWE problem. The $N \times N$ matrix represents the ciphertext, and the key is a n-dimensional vector. The addition and multiplication of ciphertext matrix are still matrices, which will not lead to the change of dimension of ciphertext calculation result. If the ciphertext matrix is "strongly B-boundary", that is, the element in ciphertext C is at most 1, and the element in error e is at most B, then a hierarchical all-homomorphic encryption scheme that can execute polynomial depth can be obtained.

2.2 Retrieval Scheme based on Homomorphic Encryption

The retrieval scheme based on homomorphic encryption proposed in this paper consists of the following five modules:

1) $Init()$. The data owner at the client side produces the public key P_k, private key S_k of the homomorphic encryption algorithm according to the parameter λ. Initializing the key E_k of document encryption algorithm.

2) $Encrypt()$. The data owner generates document vector D according to document set (DS). The document vector D is encrypted using P_k to get D_{P_k}. The document set is encrypted with E_k to get DS_{E_k}. Upload encrypted data by calling the public API interface of the application interface layer.

3) $Query()$. Date user applies P_k, S_k and E_k for date owner. When data user performs the retrieval, the original retrieval vector is expanded to the standard retrieval vector Q to get Q_{P_k} by using P_k, and the application interface layer public API is called to submit the retrieval request.

4) $Score()$. After receiving the retrieval request, Cloud Server calculates the correlation score between the retrieval vector Q and document D in the ciphertext form, and returns the retrieval vector and the correlation score of each document to the client.

5) $TopK()$. Data user decrypts the returned score with S_k, runs $TopK$ algorithm to obtain K document numbers with the highest degree of relevance, and calls the public API of the application interface layer to request document data. The server receives the request to read the encrypted document from the storage layer and returns it to the client.

2.3 Correlation Score

The similarity between the query vector and the document vector reflects the matching degree of the keyword and the document of the user query, which is the basis of the retrieval and sorting. Query vector Q and document vector D have the same dimension I, this dimension corresponds to the total number of distinct feature items in

Table 1: Symbols in this paper

λ	Security parameter
ρ	The length of the noise. In order to resist violent attack, the noise length should be taken as $\rho = w \log \lambda$.
η	The binary length of the private key. Private key length satisfies $\eta \geq \rho\Theta(\lambda \log 2\lambda)$.
τ	The number of public key. $\tau \geq \gamma + w(\log \lambda)$.
γ	The binary length of the public key. Public key length satisfies $\gamma = w(\eta 2 \log \lambda)$.

all documents. The correlation scores of query vector Q and document vector D are expressed by the vector inner product as follows:

$$D_j = (w_{1j}, w_{2j}, \cdots, w_{lj}). \quad (1)$$

$$Q_q = (w_{1q}, w_{2q}, \cdots, w_{lq}). \quad (2)$$

$$Score = sim(Q_q, D_j) = \sum_{1}^{l} w_{iq} \cdot w_{ij}. \quad (3)$$

The the correlation score between query vector Q and document vector D is higher, the document D conforms to users' query requirements sharply. If the correlation score is lower, the document D meets the user's query requirements rarely.

3 Modified Homomorphic Encryption

The basic management of cloud storage provides ciphertext retrieval function. The relevant document number can be retrieved through the query conditions given by users. The ciphertext retrieval scheme designed in this paper is based on homomorphic encryption technology. Therefore, we give a modified homomorphic encryption scheme in this paper to provide more safe retrieval.

Firstly, the symbols used in this paper are explained as Table 1.

It includes four polynomial time operations: $Setup()$, $Encrypt()$, $Circuit()$, $Decrypt()$. The proposed scheme is valid.

1) $Setup(1^n, 1^l)$. Input security parameter $n = 2^k (k \in Z)$, maximum user number l and positive integer $p \leq q = 1 mod(2n)$, q is prime number. Randomly select $s \in R_q = Z_q[x]/ < f(x) >, f(x) = x^n + 1$ as private key. Public key is $a, b = a \cdot s + p \cdot e$. In here, $a \leftarrow R_q$ is uniformly selected. Error term e is independently selected from error distribution $\chi \subset R_q$.

2) $Encrypt(id, pk, m_i)$. Given the data identification id and the user's public key (a, b). In order to encrypt a $n - bit$ plaintext message $m_i \in 0, 1^n \subset R_p$, uniform randomly select $t_i \in R_q$. Output ciphertext $(c_i^1, c_i^2) = (a \cdot t_i + pe_i^1, b \cdot t_i + pe_i^2 + m_i)$, where $e_i^j (j = 1, 2)$ is independently selected from distribution χ.

3) $Circuit(id, \alpha_i, (c_i^1, c_i^2)_{i=1}^l)$. Input identification id, ciphertext $(c_1^1, c_1^2), \cdots, c_l^1, c_l^2$ with weight $\alpha_1, \cdots, \alpha_l$. Output ciphertext $(c^1, c^2) = (\sum_{i=1}^l \alpha_i c_i^1, \sum_{i=1}^l \alpha_i c_i^2) = (\sum_{i=1}^l \alpha_i (a \times t_i + pe_i^1, \sum_{i=1}^l \alpha_i (b \times t_i + pe_i^2 + m_i)$.

4) $Decrypt(id, sk, (c_1, c_2))$. After receiving the data id, the user's private key s_k and ciphertext (c_1, c_2), the plaintext $m = \sum_{i=1}^l \alpha_i m_i$ can be obtained by calculating $(c_2 - c_1 \cdot s) \bmod p$.

Theorem 1. *To decrypt the ciphertext, suppose the ciphertext is* $c^1, c^2 = \sum_{i=1}^l \alpha_i (a \times t_i + pe_i^1, \sum_{i=1}^l \alpha_i (b \times t_i + pe_i^2 + m_i)$, *then*

$$\begin{aligned}(c^2 - c^1 \cdot s) \bmod p &= \sum_{i=1}^{l} \alpha_i (b \times t_i + pe_i^2 + m_i) \\ &\quad - s \cdot \sum_{i=1}^{l} \alpha_i (a \times t_i + pe_i^1 \bmod p) \\ &= \sum_{i=1}^{l} \alpha_i [(a \cdot s + pe) \cdot t_i + pe_i^2 + m_i] \\ &\quad - \sum_{i=1}^{l} \alpha_i \cdot s(a \cdot t_i + pe_i^1) \bmod p \\ &= \sum_{i=1}^{l} \alpha_i m_i. \end{aligned} \quad (4)$$

So the proposed scheme is proofed. The encryption scheme is linearly homomorphic.

Proof. Known message $m_i \subset 0, 1^n \subset R_p$, and weight $\alpha_i (i = 1, \cdots, l$, according to $Encrypt$ algorithm, a linear combination $\sum_{i=1}^l \alpha_i m_i$ of the message m_i has the corresponding ciphertext $(a \times t + pe^1, b \times t + pe^2 + \sum_{i=1}^l \alpha_i m_i$. On the other hand, the corresponding ciphertext of the message m_i is $(c_i^1, c_i^2) = (a \cdot t_i + pe_i^1, b \cdot t_i + pe_i^2 + m_i)(i = 1, \cdots, l)$, then ciphertext (c_i^1, c_i^2) has the corresponding linear combination $(\sum_{i=1}^l \alpha_i c_i^1, \sum_{i=1}^l \alpha_i c_i^2)$. If the $Decrypt$ algorithm decrypts the ciphertext $(\sum_{i=1}^l \alpha_i c_i^1, \sum_{i=1}^l \alpha_i c_i^2)$, the corresponding plaintext $\sum_{i=1}^l \alpha_i m_i$ can be obtained. Obviously, the corresponding plaintext of the ciphertext $(a \cdot t + pe^1, b \cdot t + pe^2 + \sum_{i=1}^l \alpha_i m_i)$ also is $\sum_{i=1}^l \alpha_i m_i$. Therefore, the encryption scheme in this paper is linear homomorphic. □

Table 2: Performance comparison with dofferent schemes

Scheme	Mould exchange	Approximate eigenvectors	Start
DGHV	None	None	$O(\lambda^{14})$
BGV	$\tilde{O}(\lambda \cdot L^3)$	None	$\tilde{O}(\lambda^2)$
Bra12	None	None	$\tilde{O}(\lambda^6)$
GSW13	None	$\tilde{O}((nL)^w)$	$\tilde{O}(n(nL)^w)$
Proposed	None	$\tilde{O}(n^w)$	$\tilde{O}(n^w)$

4 Performance Analysis of Proposed Scheme

4.1 Security Analysis

Suppose that let the advantage of a PPT adversary A to correctly distinguish its corresponding plaintext through ciphertext be ε in the chosen plaintext attack. The attack model of A is as follows:

1) $Setup(1^n)$. The challenger runs $Setup(1^n)$ to get the key $s, (a, b = a \cdot s + pe)$ and sends the public key (a, b) to the adversary A.

2) $Queries$. A random selects $m_1, \cdots, m_{q_s}$ and sends them to challenger. The challenger computes $(c_i^1, c_i^2) = Encrypt(id, (a, b), m_i)(i = 1, \cdots, q_s)$ and send it to A.

3) $Challenge$. Once the $Queries$ is completed, A outputs two different plaintext m_0 and m_1 and sends them to the challenger, the only condition being that m_0 and m_1 are not queried. It randomly selects $b \in (0, 1)$ and will challenge the ciphertext $(c^1, c^2) = (a \cdot t + pe^1, b \cdot + pe^2 + m_b)$.

4) $Output$. A outputs the guess value b' of b.

If $b' = b$, the adversary A won this game, its probability is $Pr(b = b')$, the advantages of A is $Adv = |Pr(b = b') - \frac{1}{2}|$.

4.2 Keyword Retrieval Security

The retrieval request submitted by the user to the server is the ciphertext after the conversion of P, g and x. On the premise that P, g and x cannot be obtained, the keyword plaintext M cannot be obtained, which ensures the security of the user's keyword retrieval. At the same time, the server only operates on the ciphertext in the process of performing the retrieval, and cannot learn the plaintext of user data and keywords in the whole process, thus realizing the ciphertext retrieval function.

4.3 Security of Confidential Parameters

When generating system secret parameters, it must be considered that the key P must have enough key space to prevent the key P, g and x from being exhausted. However, the larger P is, the greater the overhead is. How to achieve a good balance between efficiency and security is one of the problems that this scheme needs to focus on.

4.4 Efficiency Analysis

The retrieval efficiency of proposed scheme is closely related to the keyword length. For files with fixed length, when the plaintext M is grouped, the grouping length is larger, the grouping division will be smaller, and the corresponding retrieval cycle times will be less. On the contrary, the more groups are divided, the more cycles needed in retrieval. Meanwhile, when grouping, it is possible to divide the keyword into different groups, which leads to the failure to correctly retrieve the keyword. Therefore, the retrieval accuracy is not 100%, but decreases with the increase of the number of groups. How to design a reasonable grouping length is also one of the difficulties in this scheme. In addition, because proposed scheme supports multi-keyword joint query, it can significantly improve the retrieval efficiency compared with other schemes.

4.5 Performance Analysis

The client initializes the key and establishes the vector space model only on time, so the performance of the retrieval scheme is mainly determined by the encryption query vector, score calculation and decryption calculation TopK module. Currently, there are four classical full homomorphic encryption schemes, namely, all-homomorphic encryption scheme (DGHV) on integer, modular exchange scheme (BGV) scheme, modular invariant scheme (Bra12) scheme and approximate eigenvector scheme (GSW13) scheme. Compared result of performance with our proposed shceme is given in Table 2.

5 Comparison Results

Through the establishment of indexes with different number of keywords, the retrieval efficiency of the scheme is tested and compared with three state-of-the-art methods WMF [10], FPH [1] and OCVR [15]. Taking 100 pure Chinese text documents as test samples, the average of each document is 2MB. In the test process, the keyword length is determined to be 2 Chinese characters, corresponding

Table 3: Time comparison with three schemes

Keyword number	WMF	FPH	OCVR	Proposed method
1	11.8ms	9.6ms	8.5ms	5.4ms.
2	15.7ms	12.5ms	10.7ms	7.6ms
3	18.3ms	15.8ms	13.6ms	9.4ms
4	21.3ms	18.5ms	15.3ms	11.4ms
4	24.6ms	21.7ms	19.7ms	15.9ms

Table 4: Retrieval accuracy rate comparison with three schemes

Keyword number	WMF	FPH	OCVR	Proposed method
1	89.6%	92.3%	95.7%	98.6%.
2	84.5%	91.1%	94.8%	97.5%
3	82.1%	88.5%	89.9%	94.3%
4	80.5%	86.7%	89.2%	93.2%
4	79.4%	82.4%	87.7%	90.1%

to 32-bit binary number. The number gradually increases from 1 to 5. 50 retrieval tests are performed for each index keyword number, and the average time is taken as the final result. The result is given in Table 3.

It can be seen that the keyword retrieval efficiency of the three schemes has big difference, and proposed scheme is slightly faster. But as the number of indexes increases, the time of proposed scheme grows more slowly.

As mentioned above, keywords may be divided into different groups in ciphertext grouping, resulting in the retrieval accuracy lower than 100%. The above 100 text documents are still taken as test samples to test the retrieval accuracy of proposed method. 10 different keyword combinations are randomly selected for each quantity, 50 tests are conducted and the average accuracy is recorded. The test results are shown in Table 4.

It can be seen that with the increase of the keywords number, the accuracy of proposed scheme decreases to a certain extent, the comprehensive retrieval accuracy is above 90%.

6 Conclusions

Compared with the traditional data storage, cloud storage has the characteristics of low cost, scalability, rapid scaling and high utilization rate, which makes cloud storage get more and more attention and support. However, if the security problems in cloud storage cannot be properly solved, especially the efficient ciphertext-based retrieval problem, it will seriously restrict the sustainable development of cloud storage applications. Aiming at this urgent problem, this paper proposes a new ciphertext retrieval mechanism based on full homomorphic encryption. By double ciphertext encryption mechanism, the ciphertext retrieval can be approximately accurate without recovering the encrypted plaintext information, and the ordering of the results can be realized. This scheme not only protects the user's data security, but also improves the retrieval performance of the full homomorphic encryption algorithms in multi-keyword retrieval, which has a certain application prospect.

Acknowledgments

This work is supported by the Natural Science Fund Guiding Program in Liaoning Province (Grant No.20180520024).

References

[1] S. Bai, Y. Geng, J. Shi, *et al.*, "Privacy-preserving oriented floating-point number fully homomorphic encryption scheme," *Security and Communication Networks*, vol. 1, pp. 1-14, 2018.

[2] C. Hu, B. Yang, P. Liu, "Multi-keyword ranked searchable public-key encryption," *International Journal of Grid & Utility Computing*, vol. 6, no. 3/4, pp. 221-231, 2015.

[3] C. Lan, H. Li, S. Yin, *et al.* "A new security cloud storage data encryption scheme based on identity proxy re-encryption," *International Journal of Network Security*, vol. 19, no. 5, pp. 804-810. 2017.

[4] R. Li, Z. Dong, Y. Zhang, *et al.* "Attribute-based encryption with multi-keyword search," in *IEEE Second International Conference on Data Science in Cyberspace*, 2017. DOI: 10.1109/DSC.2017.97.

[5] H. Li, S. L. Yin, C. Zhao and L. Teng, "A proxy re-encryption scheme based on elliptic curve group," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 1, pp. 218-227, Jan. 2017.

[6] T. Lin, L. Hang, L. Jie, Y. Shoulin, "An efficient and secure Cipher-Text retrieval scheme based on

mixed homomorphic encryption and multi-attribute sorting method under cloud environment," *International Journal of Network Security*, vol. 20, no. 5, pp. 872-878, 2018.

[7] 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, 2018.

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

[9] Z. X. Lu, "Research and improvement of pageRank sort algorithm based on retrieval results," in *International Conference on Intelligent Computation Technology & Automation*, 2015. DOI: 10.1109/ICICTA.2014.119.

[10] X. Lu, J. Wang, L. Xiang, *et al.*, "An adaptive weight method for image retrieval based multi-feature fusion," *Entropy*, vol. 20, no. 8, 2018.

[11] 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, 2018.

[12] L. Teng, H. Li, "CSDK: A Chi-square distribution-kernel method for image de-noising under the IoT big data environment," *International Journal of Distributed Sensor Networks*, vol. 15, no. 5, 2019.

[13] S. Wang, K. Yan, X. Liang, "Quantitative interferometric microscopy with two dimensional Hilbert transform based phase retrieval method," *Optics Communications*, vol. 383, pp. 537-544, 2017.

[14] Y. Xu, L. Lin, H. Hu, *et al.*, "Texture-specific bag of visual words model and spatial cone matching-based method for the retrieval of focal liver lesions using multiphase contrast-enhanced CT images," *International Journal of Computer Assisted Radiology & Surgery*, vol. 13, no. 10, pp. 1-14, 2017.

[15] X. Yang, Y. Xun, S. Nepal, *et al.*, "A secure verifiable ranked choice online voting system based on homomorphic encryption," *IEEE Access*, vol. 6, no. 99, pp. 20506-20519, 2018.

[16] S. Yin, H. Li and J. Liu, "A new provable secure certificateless aggregate signcryption scheme," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 6, pp. 1274-1281, Nov. 2016.

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

[18] C. Zhang, X. Wang, J. Feng, *et al.*, "A car-face region-based image retrieval method with attention of SIFT features," *Multimedia Tools & Applications*, vol. 76, no. 8, pp. 1-20, 2017.

[19] L. Zou, X. Wang, S. Yin, "A data sorting and searching scheme based on distributed asymmetric searchable encryption," *International Journal of Network Security*, vol. 20, no. 3, pp. 502-508, 2018.

Biography

Xiaowei Wang biography. She is a full professor of the software college at Shenyang Normal University. Her interests are wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Prof. Wang had published more than 10 international journal and international conference papers on the above research fields. Email:hsiaoweiw@163.com.

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

Hang Li biography. He obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hang Li is a full professor of the software college at Shenyang Normal University. His interests are wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Prof. Li had published more than 30 international journal and international conference papers on the above research fields. Email:lihangsoft@163.com.

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

Shahid Karim biography. Shahid Karim received his BS degree in electronics from Comsats Institute of Information Technology, Abbottabad, Pakistan, and his MS degree in electronics and information engineering from Xi'an Jiaotong University, China, in 2010 and 2015, respectively. He received his PhD at the Department of Information and Communication Engineering, School of Electronics and Information Engineering, Harbin Institute of Technology (HIT), China. Now, he is an Assistant associate professor in Department of Computer Science, ILMA University. His current research interests include image processing, object detection, and classification toward remote sensing imagery. Email: shahidhit@yahoo.com.

Constructions of Balanced Quaternary Sequences of Even Length

Jinfeng Chong[1,2] and Zepeng Zhuo[1]
(Corresponding author: Zepeng Zhuo)
School of Mathematical Sciences, Huaibei Normal University[1]
Information College, Huaibei Normal Universtiy[2]
Huaibei, Anhui 235000, China
(Email: zzp781021@sohu.com)
(Received Apr. 29, 2019; Revised and Accepted Oct. 4, 2019; First Online Jan. 29, 2020)

Abstract

Pseudorandom sequences with good autocorrelation properties have been widely used in communications, cryptography, and other digital systems. In this paper, for odd prime N, new balanced quaternary sequence of even period $2N$ is constructed using Gray mapping. The distribution of autocorrelation function of the proposed quaternary sequence is also derived. Specially, a new class of balanced quaternary sequence with optimal autocorrelation value is obtained under certain condition.

Keywords: Autocorrelation Function; Balance; Binary Sequence; Quaternary Sequence

1 Introduction

Binary and quaternary sequences have received a lot of attention since they are easy to be implemented as multiple-access sequences in practical communication systems, radar, and cryptography [4]. In the application of the various wireless communication systems, the periodic autocorrelation property is used to extract desired information from the received signals. Therefore, the employed sequences should have out-of-phase autocorrelation values as low as possible to reduce interference and noise. There have been numerous researches on binary sequences with good autocorrelation property [2,8,11–14], which include m-sequence, GMW sequences, and sequences from the images of polynomials, etc. The quaternary sequences with good autocorrelation property have been also studied in [1,3,9,10].

Meng and Yan [7] proposed two constructions of binary interleaved sequences of period $4N$ by selecting appropriate shift sequences, subsequences and complement sequences, and gave their autocorrelation functions. Zhang and Yan [15] discussed the linear complexity of two classes of binary interleaved sequences given in [7].

In this paper, we present a new family of balanced quaternary sequences of even length using the inverse Gray mapping. Section 2 introduces some related definitions which would be used later. In Section 3, we give a new construction of balanced quaternary sequence, and compute the complete autocorrelation distributions of this sequence.

2 Preliminaries

2.1 Correlation Function and Difference Function

Let $\mathbf{a} = (a(t), t = 0, 1, \cdots, N-1)$ and $\mathbf{b} = (b(t), t = 0, 1, \cdots, N-1)$ be two sequences of period N over $\mathbf{Z}_m = \{0, 1, \cdots, m-1\}$. They are called $N-$ *periodic* $m-$ *ary sequences*. Let $|S|$ denote the cardinality of the set S, and define $N_k = |\{0 \le t \le N-1 : a(t) = k\}|, 0 \le k \le m-1$. If $|N_k - N_s| \le 1$ for any $0 \le k \ne s \le m-1$, then such $\mathbf{a}$ is called a balanced sequence. If there is no integer τ such that $b(t) = a(t+\tau)$ for all t, they are said to be *cyclically distinct*. The (*periodic*) *cross correlation* function between $\mathbf{a}$ and $\mathbf{b}$ at the shift $0 \le \tau \le N-1$ is defined as

$$\mathcal{R}_{\mathbf{a},\,\mathbf{b}}(\tau) = \sum_{t=0}^{N-1} \omega_m^{a(t)-b(t+\tau)},$$

where $\omega_m = e^{\frac{2\pi\sqrt{-1}}{m}}$ is the complex primitive mth root of unity.

If $\mathbf{a} = L^\tau(\mathbf{b})$ for an integer $0 \le \tau \le N-1$, where L is the left cyclic shift operator, *i.e.*, $L^\tau(\mathbf{b}) = (b(\tau), b(\tau+1), \cdots, b(\tau+N-1))$ in which the addition $t+\tau$ is performed modulo N, $\mathcal{R}_{\mathbf{a},\,\mathbf{b}}(\tau)$ is called the (*periodic*) *autocorrelation* function of $\mathbf{a}$, denoted by $\mathcal{R}_{\mathbf{a}}(\tau)$. Furthermore, these $\mathcal{R}_{\mathbf{a}}(\tau), \tau \in \{1, 2, \cdots, N-1\}$, are called the out-of-phase autocorrelation values of the sequence $\mathbf{a}$.

Let $\mathbf{a} = (a(t))$ be a binary sequence of period N. The set

$$C_{\mathbf{a}} = \{0 \le t \le N-1 : a(t) = 1\}$$

is called the *support* of $\mathbf{a}$; and $\mathbf{a}$ is referred to as the *characteristic sequence* of $C_{\mathbf{a}} \subseteq \mathbf{Z}_N = \{0, 1, \cdots, N-1\}$. For any subset A of $\mathbf{Z}_N$, the *difference function* of A is defined as

$$d_A(\tau) = |(\tau + A) \cap A|$$

for any nonzero element τ of $\mathbf{Z}_N$, where $\tau + A = \{\tau + a : a \in A\}$. Let $\mathbf{a} = (a(t))$ be the characteristic sequence of $C_{\mathbf{a}} \subseteq \mathbf{Z}_N$. It is easy to show that

$$\mathcal{R}_{\mathbf{a}}(\tau) = N - 4(|C_{\mathbf{a}}| - d_{C_{\mathbf{a}}}(\tau)).$$

or

$$\mathcal{R}_{\mathbf{a}}(\tau) \equiv N \pmod 4.$$

According to the remainder of N modulo 4, the optimal values of out-of-phase autocorrelations of binary sequences are classified into four types as follows:

1) $\mathcal{R}_{\mathbf{a}}(\tau) = \{0, \pm 4\}$, if $N \equiv 0 \pmod 4$;
2) $\mathcal{R}_{\mathbf{a}}(\tau) = \{1, -3\}$, if $N \equiv 1 \pmod 4$;
3) $\mathcal{R}_{\mathbf{a}}(\tau) = \{\pm 2\}$, if $N \equiv 2 \pmod 4$;
4) $\mathcal{R}_{\mathbf{a}}(\tau) = -1$, if $N \equiv 3 \pmod 4$, where $0 < \tau < N$.

In the first case, $\mathcal{R}_{\mathbf{a}}(\tau)$ is three level, and it can also be called optimal autocorrelation magnitude. In the last case, $\mathcal{R}_{\mathbf{a}}(\tau)$ is often called ideal autocorrelation, then binary sequence $\mathbf{a}$ of this case is called ideal sequence.

2.2 Interleaved Structure

In [5], Gong introduced the interleaved structure of sequences. Let $\{\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{T-1}\}$ be a set of T sequences of period $N, \mathbf{a}_i = (a_i(t), t = 0, 1, \cdots, N-1), 0 \le i \le T-1$. An $N \times T$ matrix is formed by placing the sequence $\mathbf{a}_i$ on the the ith column, $0 \le i \le T-1$, *i.e.*,

$$U = [\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{T-1}].$$

Concatenating the successive rows of matrix U, one can obtain an interleaved sequence $\mathbf{u}$ of period NT. For short, we write the interleaved sequence $\mathbf{u}$ as

$$\mathbf{u} = \mathcal{I}(\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{T-1}),$$

where $\mathcal{I}$ is the interleaving operator, and call $\{\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{T-1}\}$ the column sequences of $\mathbf{u}$.

Let

$$\mathbf{v} = \mathcal{I}(\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_{T-1})$$

be another interleaved sequence constructed from the column sequences $\{\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_{T-1}\}, \mathbf{b}_i = (b_i(t), t = 0, 1, \cdots, N-1), 0 \le i \le T-1$. Consider its left cyclical shift version $L^{\tau}(\mathbf{v})$, where $\tau = T\tau_1 + \tau_2 (0 \le \tau_1 \le N-1, 0 \le \tau_2 \le T-1$. It was shown that is just another interleaved sequence [5]. Namely, we have

$$L^{\tau}(\mathbf{v}) = \mathcal{I}(L^{\tau_1}(\mathbf{b}_{\tau_2}), \cdots, L^{\tau_1}(\mathbf{b}_{T-1})), L^{\tau_1+1}(\mathbf{b}_0), \cdots, L^{\tau_1+1}(\mathbf{b}_{\tau_2-1})).$$

Then, the correlation function between the interleaved sequences and at the shift becomes the summation of the inner products between the pairwise column sequences in $\mathbf{u}$ and $\mathbf{v}$, *i.e.*,

$$\mathcal{R}_{\mathbf{u},\,\mathbf{v}}(\tau) = \sum_{i=0}^{T-1-\tau_2} \mathcal{R}_{\mathbf{a}_i, \mathbf{b}_{i+\tau_2}}(\tau_1) + \sum_{i=T-\tau_2}^{T-1} \mathcal{R}_{\mathbf{a}_i, \mathbf{b}_{i+\tau_2-T}}(\tau_1 + 1). \quad (1)$$

2.3 Gray Mapping and Its Inverse

The Gray mapping $\phi : \mathbf{Z}_4 \to \mathbf{Z}_2 \times \mathbf{Z}_2$ is defined as

$$\phi(0) = (0,0), \phi(1) = (0,1), \phi(2) = (1,1), \phi(3) = (1,0).$$

Using the inverse Gray mapping $\psi : \mathbf{Z}_2 \times \mathbf{Z}_2 \to \mathbf{Z}_4$, *i.e.*,

$$\psi(0,0) = 0, \psi(0,1) = 1, \psi(1,1) = 2, \psi(1,0) = 3,$$

any quaternary sequence $\mathbf{u} = (u(t), t = 0, 1, \cdots, N-1)$ can be obtained from two binary sequences $\mathbf{a} = (a(t), t = 0, 1, \cdots, N-1)$ and $\mathbf{b} = (b(t), t = 0, 1, \cdots, N-1)$ of the same period N as follows:

$$u(t) = \psi(a(t), b(t)), 0 \le t \le N-1. \quad (2)$$

It is easily checked that

$$\omega^{u(t)} = \frac{1}{2}(1+\omega)(-1)^{a(t)} + \frac{1}{2}(1+\omega)(-1)^{b(t)}, 0 \le t \le N-1,$$

where $\omega =: \omega_4 = \sqrt{-1}$.

Krone and Sarwate [6] derived the relation between the periodic correlation function of two quaternary sequences in Equation (2) in terms of the cross correlation functions between their binary component sequences in the following lemma.

Lemma 1. *[6] Let $\boldsymbol{a} = (a(t))$ and $\boldsymbol{b} = (b(t))$ be binary sequences of period N. Then the periodic autocorrelation function $\mathcal{R}_{\boldsymbol{u}}(\tau)$ is given by*

$$\mathcal{R}_{\boldsymbol{u}}(\tau) = \frac{1}{2}[\mathcal{R}_{\boldsymbol{a}}(\tau) + \mathcal{R}_{\boldsymbol{b}}(\tau)] + \frac{\omega}{2}[\mathcal{R}_{\boldsymbol{a},\,\boldsymbol{b}}(\tau) - \mathcal{R}_{\boldsymbol{b},\,\boldsymbol{a}}(\tau)]. \quad (3)$$

2.4 Subsequence

Lemma 2. *[7] Let N be an odd number, $\boldsymbol{s} = (s(0), s(1), \cdots, s(N-1))$ be a binary sequence of period N. Take two subsequences of sequence $\boldsymbol{s}$: $\boldsymbol{s}_1 = (s(0), s(2), \cdots, s(2t), \cdots)$ and $\boldsymbol{s}_2 = (s(1), s(3), \cdots, s(2t+1), \cdots)$, where $t = 0, 1, \cdots, N-1, 2t$ and $2t+1$ are performed modulo N respectively. Then*

1) $\mathcal{R}_{\boldsymbol{s}_1}(\tau) = \mathcal{R}_{\boldsymbol{s}_2}(\tau) = \mathcal{R}_{\boldsymbol{s}}(2\tau)$;
2) $\mathcal{R}_{\boldsymbol{s}_1,\boldsymbol{s}_2}(\tau) = \mathcal{R}_{\boldsymbol{s}}(2\tau+1)$, $\mathcal{R}_{\boldsymbol{s}_2,\boldsymbol{s}_1}(\tau) = \mathcal{R}_{\boldsymbol{s}}(2\tau-1)$.

3 Constructions of balanced Quaternary Sequences from Ideal Binary Sequences

Construction: Let $N \equiv 3(\text{mod } 4), \mathbf{s} = (s(0), s(1), \cdots, s(N-1))$ be a binary ideal autocorrelation sequence of period N. Take two subsequences of sequence $\mathbf{s}$: $\mathbf{s}_1 = (s(0), s(2), \cdots, s(2t), \cdots)$ and $\mathbf{s}_2 = (s(1), s(3), \cdots, s(2t+1), \cdots)$, where $t = 0, 1, \cdots, N-1, 2t$ and $2t+1$ are performed modulo N respectively. The construction consists of the following two steps:

Step 1: Generate two binary sequences **a** and **b** of period $2N$ as

$$\mathbf{a} = I(\mathbf{s}_1, L^d(\overline{\mathbf{s}_1})), \quad \mathbf{b} = I(\mathbf{s}_2, L^d(\overline{\mathbf{s}_2}) + 1),$$

where $\overline{\mathbf{s}_{1,2}}$ are the complement sequences of $\mathbf{s}_{1,2}, L^d(\mathbf{b}) + 1 = (b(d)+1, b(d+1)+1, \cdots, b(d+N-1)+1), 1 < d < N$, and $d \neq \frac{N+1}{2}$ is an integer.

Step 2: Construct a quaternary sequence **u** of period $2N$ as

$$\mathbf{u} = \psi(\mathbf{a}, \mathbf{b}), \tag{4}$$

where ψ is the inverse Gray mapping.

Theorem 1. *The sequence* $\boldsymbol{u}$ *constructed by Equation (4) is a balanced quaternary sequence of period* $2N$*, and*

$$\mathcal{R}_{\boldsymbol{u}}(\tau) = \begin{cases} 2N, & \textit{once}, \\ -2, & N-1 \textit{ times}, \\ 0, & N-4 \textit{ times}, \\ \frac{N+1}{2}\omega, & \textit{twice}, \\ -\frac{N+1}{2}\omega, & \textit{twice}. \end{cases}$$

Proof. By Lemmas 1, 2, let $\tau = 2\tau_1 + \tau_2$, there are two cases to discuss the autocorrelation of **u**. □

Case 1: $\tau_2 = 0, 0 < \tau_1 < N$.

$$\begin{aligned} \mathcal{R}_{\mathbf{a}}(\tau) &= \mathcal{R}_{\mathbf{a}}(2\tau_1) = \mathcal{R}_{\mathbf{s}_1}(\tau_1) + \mathcal{R}_{\mathbf{s}_1}(\tau_1) = 2\mathcal{R}_{\mathbf{s}_1}(\tau_1) \\ &= 2\mathcal{R}_{\mathbf{s}}(2\tau_1), \\ \mathcal{R}_{\mathbf{b}}(\tau) &= \mathcal{R}_{\mathbf{b}}(2\tau_1) = \mathcal{R}_{\mathbf{s}_2}(\tau_1) + \mathcal{R}_{\mathbf{s}_2}(\tau_1) = 2\mathcal{R}_{\mathbf{s}_2}(\tau_1) \\ &= 2\mathcal{R}_{\mathbf{s}}(2\tau_1). \end{aligned}$$

Since $0 < \tau_1 < N, N \equiv 3(\text{mod } 4)$, then $2\tau_1 \not\equiv 0(\text{mod } N)$. Therefore, $\mathcal{R}_{\mathbf{s}}(2\tau_1) = -1$, we have $\mathcal{R}_{\mathbf{a}}(\tau) = \mathcal{R}_{\mathbf{b}}(\tau) = -2$. Also,

$$\mathcal{R}_{\mathbf{a},\,\mathbf{b}}(\tau) = \mathcal{R}_{\mathbf{s}_1,\mathbf{s}_2}(\tau_1) - \mathcal{R}_{\mathbf{s}_1,\mathbf{s}_2}(\tau_1) = 0,$$

$$\mathcal{R}_{\mathbf{b},\,\mathbf{a}}(\tau) = \mathcal{R}_{\mathbf{s}_2,\mathbf{s}_1}(\tau_1) - \mathcal{R}_{\mathbf{s}_2,\mathbf{s}_1}(\tau_1) = 0.$$

Substituting them into (3), we obtain $\mathcal{R}_{\mathbf{u}}(\tau) = -2$.

Case 2: $\tau_2 = 1, 0 \leq \tau_1 < N$.

Using (1) to the calculation of the autocorrelations $\mathcal{R}_{\mathbf{a}}(\tau), \mathcal{R}_{\mathbf{b}}(\tau)$,

$$\begin{aligned} \mathcal{R}_{\mathbf{a}}(\tau) &= \mathcal{R}_{\mathbf{a}}(2\tau_1+1) \\ &= \mathcal{R}_{\mathbf{s}_1,\overline{\mathbf{s}_1}}(\tau_1+d) + \mathcal{R}_{\overline{\mathbf{s}_1},\mathbf{s}_1}(\tau_1+1-d) \\ &= -\mathcal{R}_{\mathbf{s}_1}(\tau_1+d) - \mathcal{R}_{\mathbf{s}_1}(\tau_1+1-d), \\ \mathcal{R}_{\mathbf{b}}(\tau) &= \mathcal{R}_{\mathbf{b}}(2\tau_1+1) \\ &= -\mathcal{R}_{\mathbf{s}_2,\overline{\mathbf{s}_2}}(\tau_1+d) - \mathcal{R}_{\overline{\mathbf{s}_2},\mathbf{s}_2}(\tau_1+1-d) \\ &= \mathcal{R}_{\mathbf{s}_2}(\tau_1+d) + \mathcal{R}_{\mathbf{s}_2}(\tau_1+1-d), \end{aligned}$$

and

$$\mathcal{R}_{\mathbf{a}}(\tau) + \mathcal{R}_{\mathbf{b}}(\tau) = 0.$$

Again applying Equation (1) to the sequences **a, b**, we have

$$\begin{aligned} \mathcal{R}_{\mathbf{a},\,\mathbf{b}}(\tau) &= \mathcal{R}_{\mathbf{a},\,\mathbf{b}}(2\tau_1+1) \\ &= -\mathcal{R}_{\mathbf{s}_1,\overline{\mathbf{s}_2}}(\tau_1+d) + \mathcal{R}_{\overline{\mathbf{s}_1},\mathbf{s}_2}(\tau_1+1-d) \\ &= \mathcal{R}_{\mathbf{s}_1,\mathbf{s}_2}(\tau_1+d) - \mathcal{R}_{\mathbf{s}_1,\mathbf{s}_2}(\tau_1+1-d) \\ &= \mathcal{R}_{\mathbf{s}}(2(\tau_1+d)+1) - \mathcal{R}_{\mathbf{s}}(2(\tau_1+1-d)+1) \\ &= \mathcal{R}_{\mathbf{s}}(2\tau_1+2d+1) - \mathcal{R}_{\mathbf{s}}(2\tau_1-2d+3), \\ \mathcal{R}_{\mathbf{b},\,\mathbf{a}}(\tau) &= \mathcal{R}_{\mathbf{b},\,\mathbf{a}}(2\tau_1+1) \\ &= \mathcal{R}_{\mathbf{s}_2,\overline{\mathbf{s}_1}}(\tau_1+d) - \mathcal{R}_{\overline{\mathbf{s}_2},\mathbf{s}_1}(\tau_1+1-d) \\ &= -\mathcal{R}_{\mathbf{s}_2,\mathbf{s}_1}(\tau_1+d) + \mathcal{R}_{\mathbf{s}_2,\mathbf{s}_1}(\tau_1+1-d) \\ &= -\mathcal{R}_{\mathbf{s}}(2(\tau_1+d)-1) + \mathcal{R}_{\mathbf{s}}(2(\tau_1+1-d)-1) \\ &= -\mathcal{R}_{\mathbf{s}}(2\tau_1+2d-1) + \mathcal{R}_{\mathbf{s}}(2\tau_1-2d+1), \end{aligned}$$

and

$$\begin{aligned} \mathcal{R}_{\mathbf{a},\,\mathbf{b}}(\tau) - \mathcal{R}_{\mathbf{b},\,\mathbf{a}}(\tau) &= \mathcal{R}_{\mathbf{s}}(2\tau_1+2d+1) \\ &\quad +\mathcal{R}_{\mathbf{s}}(2\tau_1+2d-1) \\ &\quad -\mathcal{R}_{\mathbf{s}}(2\tau_1-2d+3) \\ &\quad -\mathcal{R}_{\mathbf{s}}(2\tau_1-2d+1). \end{aligned}$$

1) If $\tau_1 = \frac{N-2d-1}{2}$, then $2\tau_1+2d+1 \equiv 0(\text{mod } N), 2\tau_1+2d-1 \not\equiv 0(\text{mod } N), 2\tau_1-2d+3 \not\equiv 0(\text{mod } N), 2\tau_1-2d+1 \not\equiv 0(\text{mod } N)$. So, $\mathcal{R}_{\mathbf{u}}(\tau) = \frac{N+1}{2}\omega$.

2) If $\tau_1 = \frac{N-2d+1}{2}$, then $2\tau_1+2d-1 \equiv 0(\text{mod } N), 2\tau_1+2d+1 \not\equiv 0(\text{mod } N), 2\tau_1-2d+3 \not\equiv 0(\text{mod } N), 2\tau_1-2d+1 \not\equiv 0(\text{mod } N)$. So, $\mathcal{R}_{\mathbf{u}}(\tau) = \frac{N+1}{2}\omega$.

3) If $\tau_1 = \frac{N+2d-3}{2}$, then $2\tau_1-2d+3 \equiv 0(\text{mod } N), 2\tau_1+2d+1 \not\equiv 0(\text{mod } N), 2\tau_1+2d-1 \not\equiv 0(\text{mod } N), 2\tau_1-2d+1 \not\equiv 0(\text{mod } N)$. So, $\mathcal{R}_{\mathbf{u}}(\tau) = -\frac{N+1}{2}\omega$.

4) If $\tau_1 = \frac{N+2d-1}{2}$, then $2\tau_1-2d+1 \equiv 0(\text{mod } N), 2\tau_1+2d+1 \not\equiv 0(\text{mod } N), 2\tau_1+2d-1 \not\equiv 0(\text{mod } N), 2\tau_1-2d+3 \not\equiv 0(\text{mod } N)$. So, $\mathcal{R}_{\mathbf{u}}(\tau) = -\frac{N+1}{2}\omega$.

5) If $\tau_1 \neq \frac{N-2d-1}{2}, \frac{N-2d+1}{2}, \frac{N+2d-3}{2}, \frac{N+2d-1}{2}$, then $2\tau_1-2d+1 \equiv 0(\text{mod } N), 2\tau_1+2d+1 \not\equiv 0(\text{mod } N), 2\tau_1+2d-1 \not\equiv 0(\text{mod } N), 2\tau_1-2d+3 \not\equiv 0(\text{mod } N), 2\tau_1-2d+1 \not\equiv 0(\text{mod } N)$. So, $\mathcal{R}_{\mathbf{u}}(\tau) = 0$.

According to the above discussions about $\mathcal{R}_{\mathbf{u}}(\tau)$.

Let us now consider the matrix expression U, A, and D of the sequences $\mathbf{u}, \mathbf{a}$, and $\mathbf{b}$, respectively,

$$\begin{aligned} U &= (U_{i,j})_{0\le i\le N-1, 0\le j\le 1}, \\ A &= (A_{i,j})_{0\le i\le N-1, 0\le j\le 1}, \\ B &= (B_{i,j})_{0\le i\le N-1, 0\le j\le 1}. \end{aligned}$$

Suppose $U_{i,0} = 0$ (respectively, $U_{i,0} = 3$) for some i with $0 \le i \le N-1$. By the Gray mapping, $(C_{i,0}, D_{i,0}) = (0,0)$ (respectively, $(C_{i,0}, D_{i,0}) = (1,0)$). From the construction of the sequences $\mathbf{a}$ and $\mathbf{b}$, we know $(C_{i+d,1}, D_{i+d,1}) = (1,0)$ (respectively, $(C_{i+d,1}, D_{i+d,1}) = (0,0)$), *i.e.*, $U_{i+d,1} = 3$ (respectively, $U_{i+d,1} = 0$), where the addition of the subscript is reduced modulo N. The converse deduction holds as well. Then, we know that $N_0(\mathbf{u}) = N_3(\mathbf{u}) = N_0(\mathbf{s})$, which is equal to $\frac{N-1}{2}$ or $\frac{N+1}{2}$, the number of occurrence of 0 in a periodic segment of the ideal sequence $\mathbf{s}$. If $U_{i,0} = 1$ (respectively, $U_{i,0} = 2$) for some i with $0 \le i \le N-1$, similar arguments will lead to $N_1(\mathbf{u}) = N_2(\mathbf{u}) = N_1(\mathbf{s})$, which is equal to $\frac{N+1}{2}$ or $\frac{N-1}{2}$. Therefore, the sequence $\mathbf{u}$ is balanced, then the desired results follow. □

Example 1. *Let* $\boldsymbol{s} = (0,0,1,0,1,1,1)$ *be the binary sequence of period 7 and* $d = 2$*, then*

$$\boldsymbol{s}_1 = (0,1,1,1,0,0,1), \quad \boldsymbol{s}_2 = (0,0,1,0,1,1,1),$$

and

$$L^2(\overline{\boldsymbol{s}_1}) = (0,0,1,1,0,1,0), \quad L^2(\overline{\boldsymbol{s}_2})+1 = (1,0,1,1,1,0,0).$$

According to Construction, a balanced quaternary sequence of period 14 is generated as

$$\mathbf{u} = (0,1,3,0,2,2,3,2,1,1,1,3,2,0).$$

It is easily calculated that

$$\mathcal{R}_{\mathbf{u}}(\tau) = \{14, 0, -2, 4i, -2, 4i, -2, 0, -2, -4i, -2, -4i, -2, 0\}.$$

Remark 1. *Let* $d = 1$*. Then* $2\tau_1 - 2d + 3 \equiv 2\tau_1 + 2d - 1 (\mathrm{mod}\ N)$*. So in cases (2-2) and (2-3),* $\mathcal{R}_{\boldsymbol{u}}(\tau) = 0$*, we have the following corollary.*

Corollary 1. *Let* $0 \le \tau < 2N$*, and* $d = 1$*. The sequence* $\boldsymbol{u}$ *defined by (4) is a balanced quaternary sequence of period* $2N$*, and*

$$\mathcal{R}_{\boldsymbol{u}}(\tau) = \begin{cases} 2N, & \textit{once}, \\ -2, & N-1 \textit{ times}, \\ 0, & N-2 \textit{ times}, \\ \frac{N+1}{2}\omega, & \textit{once}, \\ -\frac{N+1}{2}\omega, & \textit{once}. \end{cases}$$

Example 2. *Let* $\boldsymbol{s}$ *be the binary sequence of period 7 defined in Example 1 and* $d = 1$*, then*

$$L^1(\overline{\boldsymbol{s}_1}) = (0,0,0,1,1,0,1), \quad L^1(\overline{\boldsymbol{s}_2})+1 = (0,1,0,1,1,1,0).$$

According to Construction, a balanced quaternary sequence of period 14 is generated as

$$\mathbf{u} = (0,0,3,1,2,0,3,2,1,2,1,1,2,3).$$

It is easily calculated that

$$\mathcal{R}_{\mathbf{u}}(\tau) = \{14, 0, -2, 0, -2, 4i, -2, 0, -2, -4i, -2, 0, -2, 0\}.$$

Remark 2. *Let* $d = \frac{N+1}{2}$*. Then* $2\tau_1 + 2d + 1 \equiv 2\tau_1 - 2d + 3 (\mathrm{mod}\ N), 2\tau_1 + 2d - 1 \equiv 2\tau_1 - 2d + 1 (\mathrm{mod}\ N)$*. So in Case 2,* $\mathcal{R}_{\boldsymbol{u}}(\tau) = 0$*, we have the following result.*

Corollary 2. *Let* $0 \le \tau < 2N$*, and* $d = \frac{N+1}{2}$*. The sequence* $\boldsymbol{u}$ *defined by (4) is a balanced quaternary sequence of period* $2N$ *with optimal autocorrelation value, and*

$$\mathcal{R}_{\boldsymbol{u}}(\tau) = \begin{cases} 2N, & \textit{once}, \\ -2, & N-1 \textit{ times}, \\ 0, & N \textit{ times}. \end{cases}$$

Example 3. *Let* $\boldsymbol{s}$ *be the binary sequence of period 7 defined Example 1 and* $d = 4$*, then*

$$L^4(\overline{\boldsymbol{s}_1}) = (1,1,0,1,0,0,0), \quad L^4(\overline{\boldsymbol{s}_2})+1 = (1,1,1,0,0,1,0).$$

According to Construction, a balanced quaternary sequence of period 14 is generated as

$$\mathbf{u} = (0,2,3,2,2,1,3,3,1,0,1,1,2,0).$$

It is easily checked that $\mathbf{u}$ has optimal autocorrelation value

$$\mathcal{R}_{\mathbf{u}}(\tau) = \{14, 0, -2, 0, -2, 0, -2, 0, -2, 0, -2, 0, -2, 0\}.$$

4 Conclusion

In this paper, for odd prime N, we presented a new construction of balanced quaternary sequences of even period 2N using the inverse Gray mapping method, and gave the distribution of the autocorrelation functions. As an extension of this paper, we will consider the linear complexity of this sequence. It would be interesting to construct balanced quaternary sequences using the Gray mapping.

Acknowledgments

This study was supported by the Natural Science Foundation of Anhui Higher Education Institutions of China (No.KJ2018A0678). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] Z. Chen, and V. Edemskiy, "Linear complexity of quaternary sequences over Z_4 derived from generalized cyclotomic classes modulo $2p$," in *International Journal of Network Security*, vol. 19, no. 4, pp. 613-622, 2017.

[2] C. Ding, T. Helleseth, and K. Y. Lam, "Several classes of binary sequences with three-level autocorrelation," in *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2606-2612, 1999.

[3] V. Edemskiy and A. Ivanov, "The linear complexity of balanced quaternary sequences with optimal autocorrelation value," in *Cryptography and Communications*, vol. 7, no. 4, pp. 485-496, 2015.

[4] S. W. Golomb and G. Gong, *Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar*, 2004. ISBN: 0521821045.

[5] G. Gong, "Theory and applications of q-ary interleaved sequences," in *IEEE Transactions on Information Theory*, vol. 41, no. 20, pp. 400-411, 1995.

[6] S. M. Krone and D. V. Sarwate, "Quadriphase sequences for spread spectrum multiple-access communication," in *IEEE Transformations on Information Theory*, vol. 30, no. 3, pp. 520-529, May 1984.

[7] R. Meng, and T. Yan, "New constructions of binary interleaved sequences with low autocorrelation," in *International Journal of Network Security*, vol. 19, no. 4, pp. 546-550, 2017.

[8] W. Su, Y. Yang, and C. Fan, "New optimal binary sequences with period $4p$ via interleaving Ding-Helleseth-Lam sequences," in *Designs Codes and Cryptography*, 2017. DOI: 10.1007/s10623-017-0398-5.

[9] W. Su, Y. Yang, Z. Zhou, *et al.*, "New quaternay sequences of even length with optimal autocorrelation," in *Science China Information Sciences*, vol. 61, no. 2, pp. 022308, 2018. DOI: 10.1007/s11462-016-9087-2.

[10] X. H. Tang and C. Ding, "New classes of balanced quaternary and almost balanced binary sequences with optimal autocorrelation value," in *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6398-6405, 2010.

[11] X. H. Tang and G. Gong, "New constructions of binary sequences with optimal autocorrelation value/magnitude," in *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1278-1286, 2010.

[12] H. Xiong, L. Qu, and C. Li, "2-adic complexity of binary sequences with interleaved structure," in *Finite Fields and Their Applications*, vol. 33, pp. 14-28, 2015.

[13] T. Yan, "New binary sequences of period pq with low values of correlation and large linear complexity," in *International Journal of Network Security*, vol. 10, no. 3, pp. 185-189, 2008.

[14] Y. Yang and X. H. Tang, "Generic construction of binary sequences of period $2N$ with optimal odd correlation magnitude based on quaternary sequences of odd period N," in *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 384-392, 2018.

[15] S. Zhang, T. Yan, Y. Sun, *et al.*, "Linear complexity of two classes of binary interleaved sequences with low autocorrelation," in *International Journal of Network Security*, vol. 18, no. 2, pp. 244-249, 2019.

Biography

Jinfeng Chong was born in 1979. She received the M.S.degree from Huaibei Normal University in 2007. Since 2002, she has been with the School of Mathematical Science, Huaibei Normal Universtiy, where she is currently an associate professor. Her research interests include cryptography and information theory.

Zepeng Zhuo was born in 1978. He received the M.S.degree from Huaibei Normal University in 2007, and the Ph.D. degree from Xidian University in 2012. Since 2002, he has been with the School of Mathematical Science, Huaibei Normal Universtiy, where he is currently a professor. His research interests include cryptography and information theory.

Research on Intrusion Detection Method Based on Hierarchical Self-convergence PCA-OCSVM Algorithm

Yanpeng Cui, Zichuan Jin, and Jianwei Hu
(Corresponding author: Zichuan Jin)

College of Network and Information Security, Xidian University
North Campus of Xidian University, Xi'an 710071, China
(Email: 546720018@qq.com)

(Received July 2, 2019; Revised and Accepted Dec. 28, 2019; First Online Apr. 6, 2020)

Abstract

At present, traditional intrusion detection methods have some shortcomings, such as long detection time, low detection accuracy and poor classification effect. This paper will combine PCA and OCSVM algorithm to build a multi-level intrusion detection model, using attack feature analysis method to preprocess data, while data cleaning and data feature selection of training set. It highlights the characteristics of abnormal data and normal data, and weakens the influence of irrelevant features on training model. PCA algorithm is used to process data to improve detection rate and reduce noise. Different models are trained by different data features to detect four attack types, namely Probe, DDOS, R2L and U2R. The optimal dimension of PCA is automatically obtained by calculating the contribution rate M of feature, which improves the traditional method that requires frequent input of K value. The model is trained by using OCSVM algorithm based on RBF core, and the disadvantage of poor classification effect of OCSVM algorithm is eliminated through improved multi-layer detection mechanism. Finally, the KDDCUP99 data set is used for experimental verification. The results show that the proposed method has more advantages than the traditional detection method.

Keywords: Intrusion Detection; KDDCUP99; Self-Convergent PCA-OCSVM

1 Introduction

With the rapid development of computer technology, people pay more attention to information security. According to CNCERT 2018 overview of China's Internet network security situation [11], it is found that there are more serious apt attacks, data leakage, distributed denial of service attacks in 2018. In 2018, CNCERT handled 106000 network security incidents. The main types of security incidents are system vulnerability exploitation and DDOS attacks. Through the combination of basic telecom enterprises and cloud service providers, the DDOS attacks launched in 2018 fell 46% year-on-year, and the accused end fell 37% year-on-year. The defense measures of security experts have played a certain role, but there are still about 20000 government websites implanted in the back door. Therefore, while strengthening the network defense means, the research of intrusion detection also needs to continue in-depth. At present, the traditional intrusion detection method mainly relies on the regular matching method to analyze the structured data stored in the database. For example, the success of Snort [23] and other intrusion detection systems is based on strong prior knowledge and customized attack rule set, but it is difficult to effectively detect unknown attacks. In addition, when the Snort Intrusion detection system matches too many or too complex rule sets, it will have a great impact on the performance of the server itself, reduce the detection rate, and even lead to the collapse of the intrusion detection system. With the rise of the field of artificial intelligence, researchers found that most of the machine learning algorithms can be applied to the field of intrusion detection with appropriate changes based on their mathematical principles [6,12,16,26].Different machine learning algorithms can achieve better results by combining with other algorithms. Intrusion detection by machine learning can reduce the workload of manual data analysis, and find more differences between abnormal data and normal data in the way of digital characteristics. Combined with the big data analysis method, according to the network traffic and the information brought by the log, we can explore the deeper correlation within the security events. To realize intrusion detection methods with higher detection rate, higher accuracy and more types of detection attacks [5].

The one class SVM studied in this paper is a classification of SVM algorithm. In [17] schölkopf and others proposed a class of SVM (one class SVM) algorithm. Its

main principle is to train data set by support vector machine, separate data points from the feature space of the origin, and maximize the distance from the hypersphere to the origin. According to different probability density, a hypersphere is divided, and the data in the area of small probability density is divided into abnormal data. One class SVM usually needs kernel function to solve nonlinear problems. Kernel function can make vector calculate inner product directly in the original low dimensional space, avoiding the complex calculation directly in the high dimensional space. Common kernel functions include linear kernel function, polynomial kernel function, Gaussian kernel function, etc. Among them, Gauss kernel is very flexible and one of the most widely used kernel functions. When using Gaussian kernel function, the choice of its parameters will have a great influence on the formation of hypersphere. In this paper, the principal component analysis (PCA) is improved. By calculating the characteristic contribution rate M, the optimal dimension of PCA is automatically obtained, and the traditional method which needs frequent input of K value is improved. Using the improved PCA algorithm to reduce the dimension and noise of the data set, make the hypersphere generated by one class SVM smooth enough, and reduce the impact of noise points on the hypersphere [19].Finally, the KDDCUP99 data set was used. The data set was produced in 1999 by DARPA, an intrusion detection evaluation project in MIT Lincoln Laboratory. Although the data set was collected in the attack log 20 years ago, it still has important reference significance for the characteristics of current network attacks. At present, although there are many changes in the means of network attack, the traffic caused by the attack is still similar to the information characteristics recorded in the log [21].
The main contributions of this paper are as follows:

1) By using the attack feature analysis method to pre filter the data, at the same time, the training set is cleaned, selected, digitized and normalized. Highlight the characteristics of abnormal data and normal data, and weaken the influence of irrelevant features on training model;

2) The PCA algorithm is improved, and the best dimension of PCA is obtained automatically by calculating its characteristic contribution rate M, which improves the traditional method that needs frequent input of K value;

3) By combining the characteristics of PCA algorithm and OCSVM algorithm, a layered PCA-OCSVM algorithm detection framework is proposed to optimize the detection model;

2 Related Research

At present, the research on SVM algorithm is still hot. Yingchao Xiao *et al.* [24] proposed to use MIES method

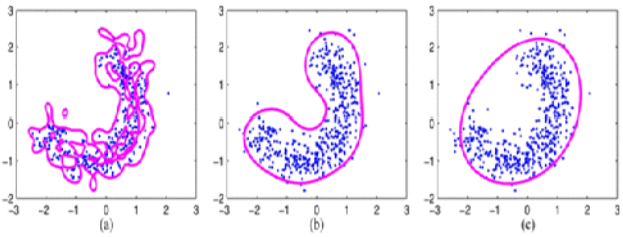


Figure 1: Different gaussian nuclear parameter results

to automatically obtain the optimal Gaussian kernel parameters, so as to obtain the optimal hypersphere. Using different Gaussian kernel parameters will have different effects on clustering results. As shown in Figure 1, this paper uses MIES method to study the geometric position of the edge and internal sample mapping in the feature space relative to the OCSVM hyperplane, and uses the distance difference between the internal sample and the edge sample to the closed surface to evaluate the applicability of the Gaussian kernel parameter D. Through the appropriate evaluation method, the optimal Gaussian kernel parameters are obtained. It can be seen that the selection of parameters of Gaussian kernel function directly affects the final classification effect. Cui Mei Bao [4] proposes to use one class SVM to implement intrusion detection based on SNMP MIB data set. Since the output of one class SVM defined in different feature spaces represents the absolute distance between the corresponding data and the decision boundary, it is not feasible to determine the related classes by comparing the absolute distances in different feature spaces. This method emphasizes the detection of DDOS. According to the protocol, DDOS is divided into tcp-syn flooding, ICMP flooding and UDP flooding. After adjusting the corresponding parameters, the classification results of DDOS attacks are better, all of them are over 98%, and the false alarm rate is still high at 9%.

Ming Zhang *et al.* [8] proposed one class SVM detection method based on Gaussian kernel function, and carried out security analysis on KDDCUP99 data set. At the same time, a new network intrusion detection model based on a class of support vector machines is proposed. The accuracy of using this model to detect normal data is as high as 100%. However, the disadvantage is that the detection rate of R2L and U2R attacks based on a class of support vector machine model is relatively low, only 26.85% and 69.23%, part of the reason that affects the accuracy of the results is the lack of data, and the establishment of R2L and U2L models is not comprehensive.

From the existing research, it can be found that OCSVM based intrusion detection has the problems of low detection rate for low-frequency attacks and single type of detection attacks [25].Therefore, this paper proposes a pca-ocsvm multi-layer detection method, which optimizes the detection effect of OCSVM multi-layer model by preprocessing and feature extraction of KDD-

CUP99 data set and smoothing with PCA algorithm.

3 Algorithm Research

3.1 Self-Convergence PCA

Principal Component Analysis (PCA) is mainly supported by covariance and covariance matrix. In signal processing, it is considered that the signal has larger variance and the noise has smaller variance. By filtering out the signal with smaller variance, the overall signal quality can be improved [10].PCA algorithm is mostly used in image processing and data dimensionality reduction. Through linear mapping, the high-dimensional data vector is projected onto the low-dimensional space, and the main components of the data are retained. That is to say, the data features with large variance are retained, and the unimportant part of data description is weakened. This can not only retain the main characteristics of data, but also reduce the amount of calculation and improve the efficiency of operation. The improved PCA algorithm flow is in Algorithm 1.

Algorithm 1 Working of the self convergence PCA

1: Begin
2: Algorithmic input: Input data set $\mathbf{X}_{mxn}$.
3: Calculate the mean $\mathbf{X}_{\text{mean}}$ of the data $\mathbf{X}_{mxn}$. set $\mathbf{X}_{\text{new}} = \mathbf{X}_{mxn} - \mathbf{X}_{\text{mean}}$
4: The calculated covariance $\mathbf{X}_{\text{new}}$ matrix is denoted as $\mathbf{X}_{\text{cov}}$,and computed eigenvalues and eigenvectors of $\mathbf{X}_{\text{cov}}$.
5: Arrange the eigenvalues from big to small, select the first k values and take the corresponding k eigenvectors as column vectors to form a matrix $\mathbf{X}_{nxk}$
6: Computing $\mathbf{X}_{\text{new}}$ $\mathbf{X}_{nxk}$,the dimension-reduced data set $\mathbf{X}_{\text{new}}$ can be obtained by projecting the matrix composed of the data set to the matrix composed of the selected feature vectors $\mathbf{X}_{\text{new}}$ $\mathbf{X}_{nxk}$.
7: Set the threshold value m according to the contribution value of each dimension of the reduced dimension data set,and round off the dimension that does not reach the threshold value, so that the number of remaining dimensions is p;
8: **while** the contribution rate of a certain dimension is less than m **do**
9: Let $k = p$ and return to step 3 until all dimension contribution values greater than or equal to m.
10: **end while**
11: End

The main components of data set screened by PCA algorithm have the following properties:

1) The principal components are orthogonal, and the difference is more significant;

2) The variance of principal components decreases in turn;

3) The data characteristics after processing lose their original explanatory nature;

4) The total amount of information remains unchanged.

Using PCA algorithm to preprocess data characteristics can highlight the internal differences of data characteristics, reduce the data processing dimension and maximize the characteristics of normal data and abnormal data, which is conducive to further exception analysis of subsequent algorithms [27]. When dealing with data sets with higher dimensions and there is a certain correlation between dimensions, PCA can be used to recombine attributes into uncorrelated principal components to represent the original information. Using PCA algorithm can also effectively reduce the dimension of sample set and improve the efficiency of operation.

3.2 OCSVM

Schölkopf *et al.* [17] extended the original SVM algorithm and proposed OCSVM algorithm. Its core idea is to transform a classification problem into a binary classification problem through hypersphere. Based on known input data sets $D = \{\mathrm{x}_i\}$,$\mathrm{x} \in \mathrm{R}^N$,$1 \leq i \leq n$.At the same time, it is assumed that there is a mapping χ from original space R^N to multidimensional space φ,and $\varphi(x_i) \in \chi$.At this point, the problem is transformed into finding a binary classifier, which divides the high-density region containing most of the normal sample points into some anomalous discrete points, which are recorded as '+1' and '-1'.

In this paper, we mainly use the Gauss kernel as the kernel function of OCSVM. For the Gauss kernel function, there are:

$$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{s}),$$
$$\langle \varphi(x_i)\ ,\ \varphi(x_j) \rangle = K(x_i, x_j) = 1.$$

It can be found that the training samples are mapped to the feature space and distributed on the circle with coordinate origin as the center and radius $R = 1$. Gaussian kernels can effectively avoid the impact of data standardization and bring a very smooth estimation to optimize the classification effect. Gaussian kernels can adjust the fitting degree by adjusting the scale parameter s.

4 Hierarchical PCA-OCSVM Model

This paper presents an anomaly detection method based on PCA-OCSVM, which integrates the characteristics of PCA and OCSVM. By extracting different data features of KDDCUP99, the original data are digitized and normalized, and then input the data set into the layered detection model [18]. Comparing the performance of OCSVM linear kernel with Gauss kernel in the algorithm, it is found that Gauss kernel has better detection effect in

Table 1: Samples of raw training and testing KDD Cup 1999 dataset

num	Example
1	0,tcp,http,SF,291,1096,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,29, 255,1.00,0.00,0.03,0.05,0.03,0.01,0.00,0.00,normal.
2	0,tcp,http,SF,219,1098,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,7, 255,1.00,0.00,0.14,0.05,0.00,0.01,0.00,0.00,normal.
3	26,tcp,ftp,SF,116,451,0,0,0,2,0,1,0,0,0,0,0,1,0,1,0,0,1,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1, 1,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,ftp_write.
4	0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,100,1,0.99,0.00,0.00,0.00,0.01,0.08,0.00,13, 245,0.23,0.15,0.23,0.25,0.00,0.00,0.08,0.00,ipsweep.
5	0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0.00,0.00,0.50,0.00,0.50,1.00,0.00,14, 245,0.29,0.14,0.29,0.25,0.00,0.00,0.07,0.00,ipsweep.

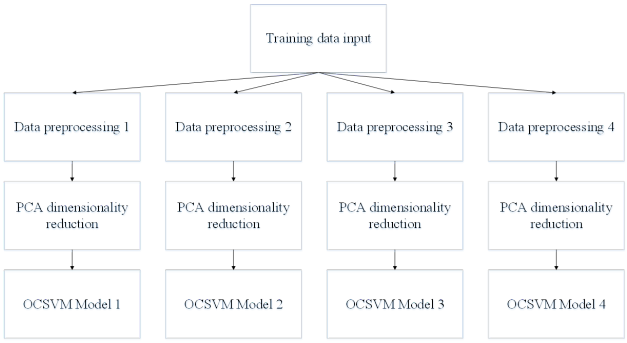


Figure 2: PCA-OCSVM training model

detection because of its superiority in dealing with nonlinear data.

4.1 PCA-OCSVM Training Method

This section mainly puts forward the training method for OCA-OCSVM model, and the method for obtaining the training model is shown in Figure 2. For the data preprocessing method in this figure, the method of digital substitution and normalization is mainly used to preprocess the KDDCUP99 data set. The original data is shown in Table 1. Some replacement methods are given in Table 2. The fields 2,3,4 columns of the dataset are digitized in a way similar to Table 2. The digitized dataset will be identified by PCA algorithm to form a matrix. The results after replacing the original data are shown in Table 3. The function of PCA algorithm is to expand the variance within the data. If the variance of some data features in the data is very large, it cannot highlight the difference between normal data and abnormal data. Therefore, it is necessary to filter and normalize the data features, otherwise the normal data and abnormal data will be confused, and the detection rate will be reduced. In the data preprocessing stage, firstly extract the data features of KDDCUP99 data set. According to the difference between U2R, L2U, Probe, DDOS attack data and normal data in different dimensions, filter the data features with large travel differences to form a new training data set, and then normalize the new data set to reduce the impact of one of the features on the data set. Expand the comprehensive impact of different dimensions of data, improve the detection accuracy. Data normalization is defined as follows:

$$X_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}$$

Table 2: Conversion table

Raw data	tcp	udp	icmp
Replacement data	1	2	3

Then PCA is used to reduce the dimension of the normal data. Finally, we use OCSVM based on Gauss kernel to train the normal model, and get the normal model in four different dimensions.

4.2 PCA-OCSVM Detection Method

In this section, a detection model of multi-layer PCA-OSVM is proposed. For the four different normal models which are trained to be used for anomaly detection of test data sets, the anomaly detection flow is shown in Figure 3.

1) The DDOS detection model is used to determine whether a DDOS attack is established or not, and the data not considered as a DDOS attack is transferred to the next model.

2) The probe detection model is used to determine whether the probe attack is valid or not, and the data not considered as the probe attack is transferred to the next model.

3) The R2L detection model is used to determine whether the R2L attack is valid or not, and the data not considered as the R2L attack is transferred to the next model.

4) U2R detection model is used to judge whether U2R attack is established or not, and data not considered as U2R attack is regarded as normal data.

Table 3: Data samples after KDDCUP99 digitization

num	Example
1	0,1,22,10,291,1096,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,29, 255,1.00,0.00,0.03,0.05,0.03,0.01,0.00,0.00,normal.
2	0,1,22,10,219,1098,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,7, 255,1.00,0.00,0.14,0.05,0.00,0.01,0.00,0.00,normal.
3	26,1,21,10,116,451,0,0,0,2,0,1,0,0,0,0,1,0,1,0,0,1,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1, 1,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,ftp_write.
4	0,3,24,10,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,100,1,0.99,0.00,0.00,0.00,0.01,0.08,0.00,13, 245,0.23,0.15,0.23,0.25,0.00,0.00,0.08,0.00,ipsweep.
5	0,3,24,10,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0.00,0.00,0.50,0.00,0.50,1.00,0.00,14, 245,0.29,0.14,0.29,0.25,0.00,0.00,0.07,0.00,ipsweep.

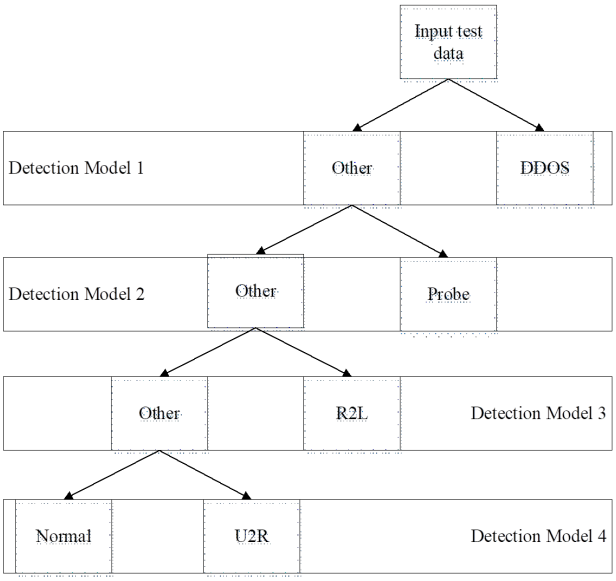


Figure 3: PCA-OCSVM detection model

OCSVM algorithm can only detect one model, and for using different kernels, the detection results will be very different. By analyzing the characteristics of KDDCUP99 data set and different kernel functions of OCSVM, this paper finds that OCSVM based on linear kernel functions performs poorly in detection results, and it is difficult to distinguish abnormal data and normal data, while using Gaussian kernel OCSVM based on different kernel parameters gets better detection results [13]. Through the layered detection mechanism, the disadvantages of OCSVM which can only detect one model are improved, and the advantages of OCSVM in solving unbalanced sample classification are continued. On the basis of making OCSVM as simple and fast as possible, it can detect many kinds of exceptions and expand the available scenarios of the algorithm.

4.3 Preprocessing of Training and Test Data

In this section, KDDCUP99 data set will be analyzed and feature selection [15], and irrelevant features in different models will be cleaned to improve the identification of detection model. The original KDDCUP99 data set contains 41 dimensional data characteristics, including the basic characteristics of TCP connection (9 kinds in total), the content characteristics of TCP connection (13 kinds in total), the time-based network traffic statistical characteristics (9 kinds in total), and the host based network traffic statistical characteristics (10 kinds in total). However, for different attack features corresponding to different attack detection models, under our proposed layered detection model, redundant features will interfere with the correct training of the model [7]. Therefore, it is necessary to pre filter the data set. According to the differences of data records caused by different kinds of attacks, the data dimensions related to the attack principle are saved and trained, and the data features irrelevant to the corresponding attack types in the 41 dimensional features are screened. For example, the data phenomenon caused by scan attack comes from the basic characteristics of TCP connection and the statistical characteristics of host network traffic. The content characteristics of the scan attack for TCP connections are roughly the same as the normal traffic, so the data samples filtered according to the scan attack characteristics are shown in Table 4. In this section, by analyzing the characteristics of Probe, DDOS, U2R and R2L attacks, different attack features are extracted to test the intrusion detection model, so as to highlight the data set differences caused by different attack modes and improve the detection accuracy [20, 22]. The data feature dimensions retained after filtering are shown in Table 5.

5 Experiments and Results

In this section, we will test the proposed layered PCA-OCSVM detection model on the VM virtual machine of Ubuntu 16.04, using 3G memory, virtual machine with 4-

Table 4: Porbe replaced data samples

num	Example
1	0,10,222,773,0,11,11,0.00,0.00,0.00,0.00,normal.
2	0,10,212,786,0,8,8,0.00,0.00,0.00,0.00,normal.
3	0,10,260,1837,0,11,11,0.00,0.00,0.00,0.00,normal.
4	1,5,0,0,0,128,2,0.00,0.00,0.53,1.00,portsweep.
5	0,6,0,0,0,17,1,0.05,1.00,0.02,0.00,ipsweep.

Table 5: Data feature dimension after filtering

Attack types	Data Feature Dimensions Retained after Screening
Probe	1,4,5,6,11,23,24,38,39,40,41
DDOS	5,6,13,23,24,25,26,29,30,32,33,34,35,37
U2R	1,3,10,11,13,14,15,16,17,18,19,23,24,25,31,32,33,34,35,36
R2L	4,10,11,14,15,16,17,18,19,23,24,27,28,32,33,36,38,39,40,41

core CPU performance and python 2.7.6 compilation environment. KDDCUP99 data set is adopted for training and test data. After analyzing contribution rate of each dimension by improved self-convergence PCA algorithm, the threshold value of contribution degree is m = 0.001. Dimension parameters under different attack models obtained by self-convergence are shown in Table 6. Table 7, Table 8, Table 9 and Table 10 show the detection results of intrusion detection test based on the parameter model obtained by self-convergence algorithm. The results show that this method can optimize the engineering efficiency, quickly obtain excellent dimensional parameters, and has a high detection accuracy. Let $j\epsilon\{Probe, DDOS, U2R, R2L\}$, i be the corresponding subclass attack under each big class attack. The accuracy is AC_{ji}, the number of tests is TQ_{ji}, the number of hits is HQ_{ji}, and the average accuracy is AAC_j. The calculation formula is as follows:

$$AC_{ji} = \frac{HQ_{ji}}{TQ_{ji}}$$

$$ACC_j = \frac{\sum_i HQ_{ji}}{\sum_i TQ_{ji}}$$

Table 6: Algorithm parameters corresponding to different models

Attack types	m	n	gamma	nu
Probe	0.001	8	1	0.1
DDOS	0.001	8	1	0.1
U2R	0.001	10	5	0.1
R2L	0.001	7	5	0.1

Through the comparison of experiments, it is found that when the parameter nu and gamma are larger, the fitting degree of OCSVM model is higher, and when the parameter nu and gamma are smaller, the fitting degree is lower. In the process of engineering implementation, increasing gamma parameter can improve the detection rate, but the false alarm rate of normal data will also increase. By reducing the nu parameter, the false alarm rate of normal data can be reduced, but the detection rate of abnormal attacks will also be reduced [1]. For example, in Table 10, R2L model has a low detection rate for guess_passwd attack, but when we choose to remove the 23, 24 witter sign, the detection rate for guess_passwd attack can reach 100%. This is because the contribution rate of 23 and 24 features is large when training the normal model, and the PCA algorithm still has a great impact on the data set features after the principal component extraction, but it can not show a good effect for guess_passwd attack detection. In fact, the removal of 23-dimensional and 24-dimensional features will also lead to a decrease in the detection rate of other types of attacks on R2L. This is because for other attacks on R2L, these two types of attacks can help the detection model to obtain a better distinction between abnormal data and normal data. In fact, the removal of 23-dimensional and 24-dimensional features will also lead to a decrease in the detection rate of other types of attacks against R2L. This is because for other subclass attacks in R2L attack type, these two features can help the detection model to obtain better differentiation between abnormal data and normal data. Therefore, it is found that for some special types of attacks, a special detection model can be established to improve the reliability of the intrusion detection system.

By comparing [1,8] with the detection model proposed in this paper, it can be found that the detection rate of the intrusion detection method proposed in this paper is similar to that of [1, 8] in the detection of Probe and DDOS attacks, but it is greatly improved in the detection of U2R and R2L attack types. The comparison results are shown in Figure 4. This shows that the Hierarchical PCA-OCSVM Model method proposed in this paper is better than OCSVM and SVM-ELM detection method in

Table 7: Detection results for different probe attacks

Probe Attack Types	Test Quantity	Accuracy Rate
Ipsweep	1247	93.5
Portsweep	1040	99.9
Nmap	232	88.8
Satan	1589	99.9
normal	5000	96.8

Table 8: Detection results for different DDOS attacks

DDOS Attack Types	Test Quantity	Accuracy Rate
teardrop	979	100
smurf	280790	99.9
pod	264	76.5
neptune	107201	99.9
normal	5000	98.5

Table 9: Detection results for different U2R attacks

U2R Attack Types	Test Quantity	Accuracy Rate
buffer_overflow	30	86.7
loadmodule	9	100
perl	3	100
rootkit	10	100
normal	5000	98.1

Table 10: Detection results for different R2L attacks

R2L Attack Types	Test Quantity	Accuracy Rate
warezclient	1020	93.5
warezmaster	20	100
multihop	7	85.7
imap	12	100
ftp_write	8	100
guess_passwd	53	30.2
normal	5000	98.7

Table 11: Detection results for different attacks

Attack types	Average Accuracy Rate
Probe	97.4
DDOS	99.9
U2R	92.3
R2L	90.6
normal	97.1

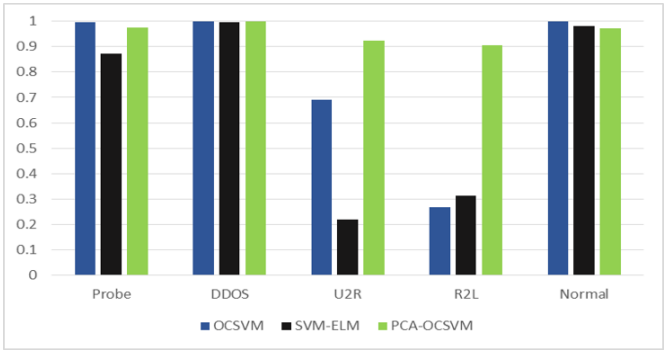


Figure 4: Comparison of hierarchical PCA-OCSVM detection accuracy with other methods

some functions, to some extent, it overcomes the problem of poor classification when OCSVM detects multiple attacks, and improves the effect of anomaly detection. The average accuracy for different attacks is shown in Table 11.

6 Conclusion

Firstly, this paper investigates the severe forms of current network security, and finds that the current network attacks can be divided into four major categories: Probe, DDOS, R2L and U2R. However, various kinds of small attack methods emerge in endlessly under different large categories, and traditional detection methods are gradually being broken down, so the intrusion detection method based on machine learning is becoming more and more important [3, 14].

According to the characteristics of OCSVM algorithm, this paper combines it with PCA algorithm to enlarge the difference between normal data and abnormal data. At the same time, the characteristics of KDDCUP99 data set are analyzed, and the data characteristics of different attack types are screened and preprocessed. The experiment of anomaly detection is carried out by using the OCSVM algorithm based on different kernel parameters. The disadvantages of OCSVM are improved by the multilayer anomaly detection model. By specifying the threshold value of the lowest characteristic contribution rate in PCA, the self-convergence function of PCA dimension parameters is realized, and the engineering implementation of PCA algorithm is optimized. Finally, based on KDDCUP99 data set, we test the accuracy of Probe, DDOS, R2L, U2R and different attack methods including these four attack types. The test results show that the detection accuracy of the proposed detection method for various types of attacks can reach 100%, and through the statistical average detection accuracy compared with previous studies, the proposed detection method is more excellent. In the follow-up research, we will focus on how to realize the automatic optimization and dynamic adjustment of parameters, so that the detection method can adapt

to different detection environment faster and get better detection effect. In the project implementation, the original KDDCUP99 data set is combined with the data set generated by the new attack [2, 9], so that the intrusion detection model can detect more kinds of network attacks.

References

[1] W. L. Al-Yaseen, Z. A. Othman, M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296–303, 2017.

[2] K. K. R. Amrita, "A hybrid intrusion detection system: Integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers," *International Journal of Network Security*, vol. 21, no. 3, pp. 438–450, 2019.

[3] K. K. R. Amrita, "Design of network threat detection and classification based on machine learning on cloud computing," *Cluster Computing*, vol. 22, no. 1, pp. 1–10, 2019.

[4] C. M. Bao, "Intrusion detection based on one-class SVM and SNMP MIB data," in *The Fifth International Conference on Information Assurance and Security*, pp. 346–349, Aug. 2009.

[5] M. Biba, L. Nishani, "Machine learning for intrusion detection in manet: A state-of-the-art survey," *Journal of Intelligent Information Systems*, vol. 46, no. 2, pp. 391–407, 2016.

[6] H. Duan, H. Hu, W. Qian, H. Ma, X. Wang, A. Zhou, "Incremental materialized view maintenance on distributed log-structured merge-tree," in *Pringer International Publishing AG, Part of Springer Nature*, pp. 682–700, 2018.

[7] L. Feng, Y. Wang, "Hybrid feature selection using component co-occurrence based feature relevance measurement," *Expert Systems with Applications*, vol. 102, pp. 83–99, 2018.

[8] J. Gong, M. Zhang, B. Xu, "An anomaly detection model based on one-class SVM to detect network intrusions," in *International Conference on Mobile Ad-hoc and Sensor Networks (MSN'16)*, pp. 102–107, 2016.

[9] A. Guezzaz, A. Asimi, Y. Asimi, Z. Tbatou, Y. Sadqi, "A global intrusion detection system using pcapsocks sniffer and multilayer perceptron classifier," *International Journal of Network Security*, vol. 21, no. 3, pp. 438–450, 2019.

[10] Z. Heng, *Design and Implementation of ELK based Network Security Log Management and Analysis System*, Beijing University of Posts and telecommunications, 2017.

[11] X. Jian, W. Xiaoqun, H. Zhihui, "Overview of china's internet security situation in 2018," *Confidential Science and Technology*, vol. 5, 2019.

[12] J. Jixue, H. Yingjie, Y. Zongmin, "Overview of machine learning application in intrusion detection," *Computer security*, no. 3, pp. 20–21, 2010.

[13] S. Khare, S. Y. Sait, A. Bhandari, "Multi-level anomaly detection: Relevance of big data analytics in networks," *Sadhana*, vol. 40, no. 6, pp. 1737–1767, 2015.

[14] Z. Liu, Y. Xin, L. Kong, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, no. 99, pp. 1–1, 2018.

[15] P. Padiya, U. Ravale, N. Marathe, "Feature selection based hybrid anomaly intrusion detection system using k-means and RBF kernel function," *Procedia Computer Science*, vol. 45, no. 39, pp. 428–435, 2015.

[16] Z. Qi, Z. Kun, "Application of machine learning in network intrusion detection," *Data Collection and Processing*, vol. 32, no. 3, pp. 479–488, 2017.

[17] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 7, pp. 1443–1471, 2001.

[18] D. Shin, D. Kim, Y. H. Kim, "Fast attack detection system using log analysis and attack tree generation," *Cluster Computing*, no. 2, pp. 1–9, 2018.

[19] J. Shouling, Q. Yaguan, L. Hongbo, "A toxic attack method for SVM based intrusion detection system," *Acta Electronica Sinica*, vol. 47, no. 1, pp. 59–65, 2019.

[20] M. Touahria, S. Maza, "Feature selection for intrusion detection using new multi-objective estimation of distribution algorithms," *Applied Intelligence*, vol. 49, no. 12, pp. 4237-4257, 2019.

[21] University of California, "Kdd cup 1999 data," *The UCI KDD Archive Information and Computer Science*, 2019. (`http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`)

[22] H. Xiangjing, H. Liang, C. Zemao, "Research on intrusion detection algorithm based on improved k-means clustering," *Computer and digital engineering*, vol. 6, pp. 1145–1149, 2017.

[23] L. Xing, *Research and Design of DoS Attack Detection System based on Snort*, Beijing University of Posts and telecommunications, 2015.

[24] W. Xu, Y. Xiao, H. Wang, "Parameter selection of gaussian kernel for one-class SVM," *IEEE Transactions on Cybernetics*, vol. 5, pp. 927–939, 2015.

[25] Z. Yan, Z. Jin, Y. Cui, "Survey of intrusion detection methods based on data mining algorithms," in *Proceedings of International Conference on Big Data Engineering*, pp. 98–106, June 2019.

[26] W. Zhao, Q. Liu, P. Li, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE Access*, vol. 6, no. 99, pp. 12103–12117, 2018.

[27] S. Zhonglin, N. Lei, "Pca-akm algorithm and its application in intrusion detection," *Computer Science*, no. 2, pp. 41, 2018.

Biography

Yanpeng Cui, born in 1978, female, doctoral student, lecturer. The main research fields are electronic warfare signal processing, machine learning, intrusion detection and so on..

Zichuan Jin, born in 1995, male, master's degree student, mainly studies machine learning, network operation and maintenance, intrusion detection and other directions.

Jianwei Hu, born in 1973, male, Ph. D. doctoral student. He mainly studies computer network, industrial control system, network hardware and software security and attack-defense confrontation.

Analyzing System Log Based on Machine Learning Model

Chia-Mei Chen, Gen-Hong Syu, and Zheng-Xun Cai
(Corresponding author: Chia-Mei Chen)

Department of Information Management, National Sun Yat-sen University
No. 70, Lianhai Road, Gushan District, Kaohsiung 804, Taiwan
(Email: cchen@mail.nsysu.edu.tw)
(Received Feb. 7, 2020; revised and accepted June 8, 2020)

Abstract

Cyberattacks become one of the most concerning threats to governments, businesses, and people. Efficient incident investigation is vital to identify the root cause of an attack, which requires expertise in analyzing audit logs. System logs are an important audit trace to understand the status of systems and useful for analyzing anomalies in case of a security breach. However, due to the diversity and massive volume of the logs, log analysis requires expertise domain knowledge as well as a large amount of manpower and computing resources. To make an incident investigation more accessible, this study proposes a machine-learning-based system log analysis approach that identifies suspicious event activities automatically. The experimental results show that the proposed neural network-based approach could identify malware efficiently with the precision of 95.5% and outperforms SVM.

Keywords: Anomaly Detection; Big Data Analysis; Machine Learning

1 Introduction

The explosive expansion of electronic commerce offers unprecedented opportunities for businesses to expand their markets. Government, businesses, and people rely on seamless ubiquitous computing services heavily, putting valuable and confidential data over the internet and on clouds. However, these convenience services have been accompanied by a commensurate increase in cyber-attacks and caused serious damages and financial losses. Cybercrimes kept increasing worldwide these years. Not only have financial firms suffered serious losses from cyberattacks, but also high-tech companies, governments, and academic institutes have experienced severe data breaches.

Attackers attempt to exploit vulnerabilities at different layers, such as most common application technical or logic attacks [12], and penetrate target systems. In order to detect suspicious behaviors, systems record important activities performed. The process of recording events during the execution of the operating system, process, network, or application is called logging, which produces log files composed of useful information associated with events that occurred in the system, network, or application [11] and is useful for analyzing anomalies in case it is compromised. Incident investigation faces enormous data collection as well as data analysis, which consumes a lot of time for system administrators or incident investigators to discover suspicious behaviors from a massive amount of system logs. An efficient system log analysis method is desired to identify suspicious events.

There are two common attack detection approaches: misuse detection and anomaly detection. Misuse detection is a rule-based approach that contains the signatures of intrusion patterns and effective in identifying known attacks but usually performs poorly on new attacks or variants of known attacks. Anomaly detection approaches profile normal or abnormal behaviors and apply deviations or similarities from the respective profile to anomaly detection, which may employ statistics, machine learning, or data mining techniques to train the detection model.

Most studies analyzed network traffic or alerts from defense systems, but the issue of analyzing system logs was not yet fully explored in the literature. In real-world cases, attackers circumvented the defense mechanisms and implanted malware into target systems, and the defense systems failed to discover them.

Identifying attacks efficiently is time-critical in order to reduce the damage caused by an attack. The more informative audit trails are, the more efficiently the detection model could build to identify attacks. System logs provide informative data about the activities performed on systems and are useful for detecting suspicious events. However, analyzing audit trails is labor-intensive, and most organizations are short of security professionals as well as resources to perform the task promptly and efficiently. An automatic and efficient system log analysis method is desired.

Over 78% of systems are Windows-based [19], and in

addition, Windows has become attackers' favorite hacking platform [20]. Event logs provided by Sysmon (System Monitor) are comprehensive and useful for identifying suspicious activities on Window-based systems. Hence, this study focuses on analyzing system logs and identifying anomalous events for Windows-based systems.

Neural Networks (NNs) are one of the most popular machine learning algorithms at present. It has been decisively proven over time that NNs outperform other algorithms in accuracy and speed [13]. Recurrent neural networks (RNNs), a variation of NNs, handle time-series data efficiently like system logs. In order to make system log analysis and anomaly detection accessible for short-of-staff organizations, this study proposes an RNN-based system log analysis approach to automatically identify suspicious behaviors.

The rest of the paper is structured as follows: Section 2 briefly reviews the related studies on attack detection and log analysis and a background study of Windows system logs; Section 3 elaborates the proposed detection; The performance evaluation is presented in Section 4 followed by the concluding remarks and future work.

2 Related Work

Raftopoulos and Dimitropoulos [16] asserted that the alerts from intrusion detection systems often produce lots of false positives. They proposed an alert reduction method that employs entropy-based information-theoretic criteria to find recurring alerts. Zargar *et al.* [23] proposed an intrusion detection framework for a distributed computing environment where service providers collaborate to cope with attacks. Lo [9] proposed a framework for detecting attacks by exchanging alert information with other intrusion detection systems. A comprehensive trust management scheme is required to support the trust relationship among the service providers.

Liu *et al.* [18] proposed an alert correlation model for constructing attack scenarios that require the given attack graphs and signature rules to correlate security events. Siraj *et al.* [1] proposed a framework of attack prediction, which includes the following components: alert normalization, reduction, prioritization, and attack scenario construction and prediction. To obtain effective detection results, Amini *et al.* [2] proposed a detection method that combines multiple classifiers: neural network, fuzzy clustering, and stacking combination methods. The experimental results showed that the proposed multi-classifier approach performed better than single classifiers.

A significant amount of research has been contributed to attack detection. Various approaches including data mining, finite state machines, etc. have been explored in order to propose efficient approaches to identifying anomalies.

Serketzis *et al.* [17] proposed a log management system that collects the audit logs from network equipment, security devices, operating systems, and applications. Users could facilitate search function for discovering suspicious events. Dwyer and Truta [6] proposed a statistic-based approach for identifying anomalies in Windows event log data using standard deviation. The average number and standard deviation of the events of a specific type at any time of a day for any server or user are calculated; an anomaly is determined if an event goes outside of the standard deviation. The results show the proposed solution lowers the amount of logs to be reviewed to a feasible amount.

A Windows event log is a detailed record of the system, security, and application notifications stored by the Windows operating system and useful for identifying system faults and attacks. It can be broadly categorized into two levels of logs: the operating system and applications. Both can utilize event logs to record important events. Windows system logs record software installation, security management, system setup operations on initial startup, and problems or errors; the other level is service logs that record application related events. To identify suspicious events on a system, this study focuses on analyzing Windows system logs.

System Monitor (Sysmon) is a Windows system service to monitor and record system activities to the Windows event logs that provide detailed information about process creations, network connections, and changes to file creation time. Sysmon does not provide event analysis, but the official document claims that, by collecting and analyzing the event logs, anomalous activities can be discovered.

Various machine learning algorithms have been developed recently based on deep learning that exhibits remarkable capabilities in classification and clustering. Among them, supervised machine learning algorithms have been applied to anomaly detection. A classification task involves separating data into categories based on the information learned from the labeled training data, where each instance in the training data set contains one "target value" (*i.e.* class label) and some "attributes" (*i.e.* features or observed variables). The goal of a supervised learning algorithm is to produce a model (based on the training data) which classifies or predicts the target values of the test data given only the test data attributes.

The supervised learning machine model, SVM (support vector machine), is widely used in classification. Given a training set of instance-label pairs (xi; yi), an SVM model is to find a linear hyperplane with the maximal margin to separate the data. Four basic types of kernel functions are used in modeling the hyperplane: Linear, polynomial, radial basis function (RBF), and sigmoid. In general, the RBF kernel is a reasonable choice. This kernel nonlinearly maps the samples into a higher-dimensional space, so it can handle the case of a nonlinear relation between class labels and attributes. The linear kernel function is a special case of the RBF, and the sigmoid kernel behaves like RBF for certain parameters.

Zidi *et al.* [24] employed SVM classification model to

identify failures in wireless sensor networks and claimed that the fault detection has to be precise to avoid negative alerts and rapid to limit loss. Comparing with other algorithms, their study indicates that SVM is efficient. Wang *et al.* [22] proposed an efficient intrusion detection framework based on SVM and concluded a similar finding that SVM achieves a better performance than the existing methods in terms of accuracy, detection rate, false alarm rate, and training speed. Anton *et al.* [3] applied SVM to network anomaly detection on industrial environments and the experimental results showed that SVM performs well.

NN is one of the most popular machine learning algorithms. Convolutional Neural Networks (CNNs) perform well on pattern recognition and image categorization but are not suitable for data with time sequence. The neural network models have been employed on anomaly detection. Radford *et al.* [15] showed that RNN can represent sequences of communications on a network and discover anomalous network traffic. Prasse *et al.* [14] analyzed HTTPS network flows, employed a natural language model to extract features from domain names, and proposed an LSTM-based detection method, where LSTM (Long Short-Term Memory) is an RNN model dealing with time-series data. Their experimental results show that the LSTM classification model outperforms a random forest model. Kim and Ho [8] employed CNN to extract spatial features and LSTM temporal characteristics and proposed a neural network for detecting anomalies on web traffic.

3 System Design

According to the literature review, attackers might customize their attack to evade the detection mechanism. Sysmon logs provide detailed information about the actions performed on a system, including network connections, running processes, registry files, and file systems. Analyzing audit logs with informative details could improve detection performance. RNN models are suitable for analyzing time sequence data like such audit logs. This study proposes an RNN-based log analysis method to automatically classify suspicious events, where the system model is plotted in Figure 1. This research collected malware behaviors by emulating attacks in a controlled environment and collected normal user behaviors from a campus network. Both parts of the labeled data were verified manually.

Our preliminary study discovered that most attacks contain the following four types of behaviors: process, file access, registry, and network access, and all can be captured by the system logs. Log records are informative, but most are benign. Therefore, the preprocess module extracts security-relevant event records from log files in order to reduce the processing time in the following steps, where the security-relevant event records extracted represent the behavior of a given process. The proposed RNN model learns to classify misbehaviors from the labeled data collected from benign users and malware. The detail of the proposed method is explained below.

3.1 Security-Related Events

To identify the misbehaviors of a process, process behaviors should be captured and analyzed. In this study, process behaviors are delineated as a sequence of the events captured by the system logs. Based on our preliminary study, the above four types of security-relevant events are selected from Sysmon event logs for identifying critical activities and status changes of a system. As event attributes provide the detailed information of an event, key event attributes are selected to improve the description of an action performed. In summary, by describing process behaviors in a sequence of the performed events with the associated event attributes, the proposed anomaly detection method plots the status changes of the system and discovers anomalous behaviors. The security-relevant events and attributes selected in the proposed method are explained below.

Process Events

A binary image file needs to be loaded into memory to be able to run this program, while some malware injects its executable file to another legitimate process or kills a process. Such behaviors can be captured by process events. Based on the above injection anomaly and the literature review on malware misbehaviors, the selected critical process events/behaviors include creating a process, terminating a process, loading image, creating a remote thread, and accessing a process, where Table 1 summarizes the selected process events and attributes.

File Access Events

Malware accesses file system for various purposes. Downloader or dropper may download additional malware or payloads to perform further attack; ransomware accesses and encrypts documents and files; some malware steals and sends out confidential information. The file access events record the access behaviors of the file system. The selected critical file access events include changing file creation time, accessing a file, creating a file, and creating a file hash, where Table 2 summarizes the selected file access events and attributes.

Registry Events

The registry [10] is a hierarchical database that contains data about the operation of the Windows operating system, applications, and services. The data is structured in a tree format. Each node in the tree is called a key. Each key can contain both subkeys and data entries called values. Sometimes, the presence of a key is all the data that an application requires; other times, an application opens a key and uses the values

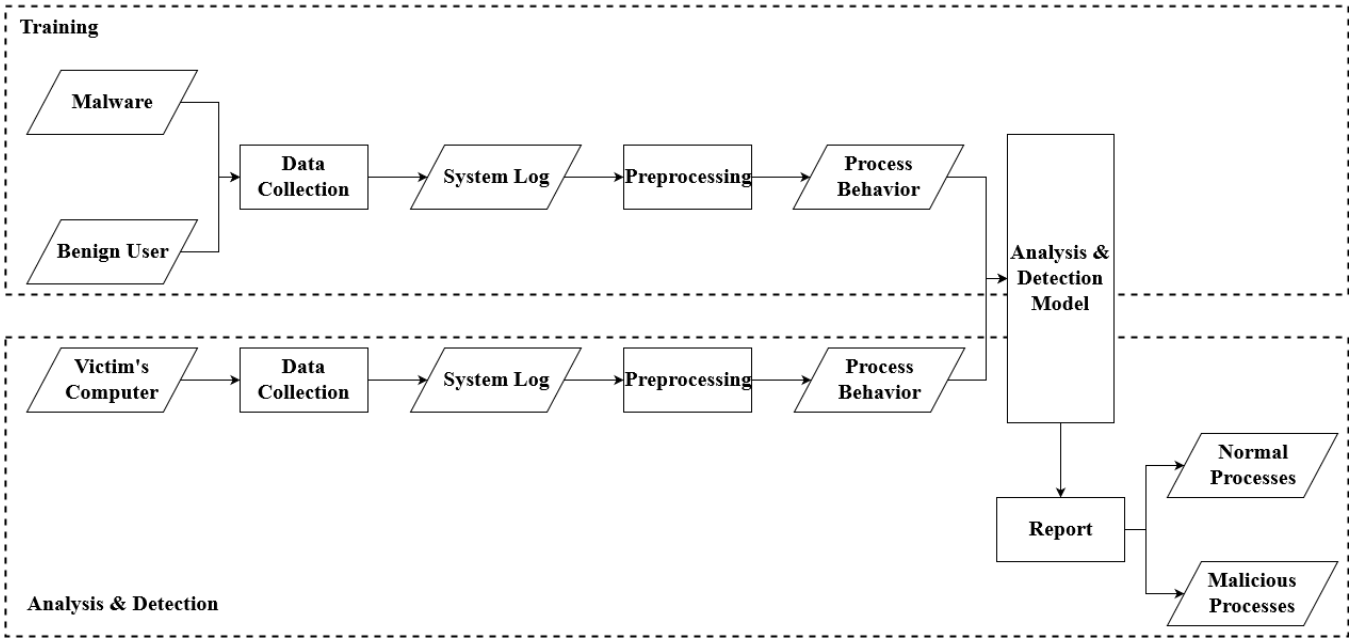


Figure 1: The proposed log analysis and anomaly detection method

associated with the key. Registry file contains important information including applications installed, files and paths recently accessed, network setup, and account information. Malware [7] might modify registry keys in order to achieve persistence on a system, like exploiting Run, RunOnce, BootExecute, Winlogon, and Startup Keys. The selected critical registry events include setting a registry key value, renaming a registry key and value, creating and deleting a registry object, where Table 3 summarizes the selected registry events and attributes.

Network Events

Most malware performs network connections for various purposes. For example, botnets connect to the command and control server for reporting victim information and receiving attack instruction; miner malware connects to the mining pool; Some malware attempts to exploit and infect more machines. Hence, network connections are critical in identifying misbehaviors. Table 4 summarizes the chosen network event and attributes.

3.2 Log Analysis and Anomaly Detection Method

The Sysmon system logs record the events performed by all the running processes in a system but lose the time order of process events. However, the time sequence is crucial to understand process behaviors. This study utilizes Process ID to link all the event records of a process in chronological order, and the observed process events are outlined in Figure 2.

The traditional neural network models perform poorly on time-series datasets; based on literature review LSTM is suitable for dealing with time-series data. This study employs an improved RNN as well as a variation of LSTM: GRN (Gated Recurrent Unit) [4],as it eliminates the vanishing gradient problem faced by standard RNN and produces equally excellent results as LSTM. The proposed GRU classification model consists of the input

Table 1: The selected process events and attributes

Event	Attribute
Event ID 1: Process creation	Event ID
	Image
	User
	ParentImage
Event ID 5: Process terminated	Event ID
Event ID 7: Image loaded	Event ID
	ImageLoaded
	Signed
Event ID 8: CreateRemoteThread	Event ID
	TargetImage
Event ID 10: ProcessAccess	Event ID
	TargetImage
	GrantedAccess

Table 2: The selected file access events and attributes

Event	Attribute
Event ID 2: A process changed a file creation time	Event ID
	TargetFilename
	CreationUtcTime
	PreviousCreationTime
Event ID 9: RawAccessRead	Event ID
Event ID 11: FileCreate	Event ID
	TargetFilename
Event ID 15: FileCreateStream-Hash	Event ID
	TargetFilename

Table 3: The selected registry events and their attributes

Event	Attribute
Event ID 12: RegistryEvent (Object create and delete)	Event ID
	EventType
Event ID 13: RegistryEvent (Value Set)	Event ID
Event ID 14: RegistryEvent (Key and Value Rename)	Event ID
	EventType

Table 4: The selected network event and attributes

Event	Attribute
Event ID 3: Network connection	Event ID
	Protocol
	Initiated
	SourcePort
	DestinationPort

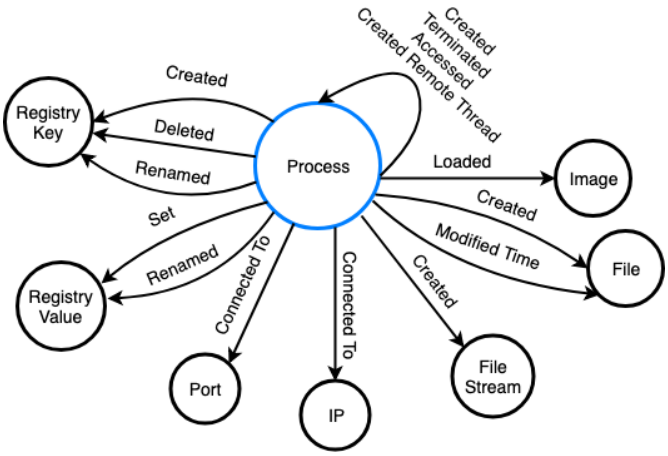


Figure 2: The events of a process

layer x, embedding layer E, GRU layer G, and output layer y, where the proposed model structure is outlined in Figure 3.

Non-numerical input data needs to be encoded in order to feed into the input layer of the proposed GRU classification model. The proposed features belong to one of the following value types: numerical, categorical, and string data. Most of the proposed features are numerical data like event ID; some belong to categorical data; attributes like file name or environment variables belong to string data. The numerical data is straightforward and does not need any encoding. The categorical data is encoded by numbers, where each category is denoted by a different number. One-hot encoding is commonly used for encoding string data but is inefficient in handling sparse data. This study employs label encoding to reduce the dimension embedding and to reduce the processing time.

The input to the input layer is composed of a sequence of the events performed by a process as described above. Let N_{events} be the maximum number of the events to be captured to represent the behavior of a process and N_{attr} be the maximum number of the associated event attributes. All time-series need to have the same length. In order words, the behavior of a process is represented as a matrix of $N_{events} \times N_{attr}$ at the input layer and zero is padded if it is needed to match the required dimension. The GRU classification model applies vector transformation on the embedding layer. The GRU layer consists of N_{events} neurons, matching with the number of the events in a process, and learns the relationship of the events and attributes among benign and malicious processes from the training data.

4 Performance Evaluation

To evaluate if the proposed method can identify malicious behaviors and unknown malware, this study emulated machines compromised by various types of malware. Each injected attack was executed for 5 minutes in a controlled environment, and the system logs were collected during the execution.

10051 malware samples from 36 different malware families were retrieved from Malware Knowledge Base hosted by the National Center for High-performance Computing, where 8889 have been analyzed by VirusTotal, and the rest had not been uploaded or analyzed at the time the evaluation conducted and were considered as unknown malware in this study. A total of 10048 event records were collected from the aforementioned experiments.

The normal behaviors with common benign program execution were collected from a campus network, including Windows system processes and common user processes such as document editing software, web browsing, and other benign applications. A total of 47175 event records were obtained from the benign.

4.1 Performance Evaluation of the Proposed Model

The proposed ML-based detection system aims for analyzing system logs and classify anomalies efficiently. This study adopts accuracy as the performance measurement as classifying benign and malicious behaviors correctly is equally important. Accuracy is expressed below, which is calculated from the confusion matrix summarized in Table 5.

Table 5: The selected network events and their attributes

	Benign (Detected)	**Malicious (Detected)**
Benign	True Negatives (TN)	False Positives (FP)
Malicious	False Negatives (FN)	True Positives (TP)

Table 6 lists the parameter settings of the proposed GRU model, where N_{events} is set to 100 and N_{attr} is set to 100 and Nattr is set to 8; 100 events are extracted to represent the behaviors of a process and the maximum of 8 event attributes from an event. The experiments on 10-fold cross-validation with different ratios (training:

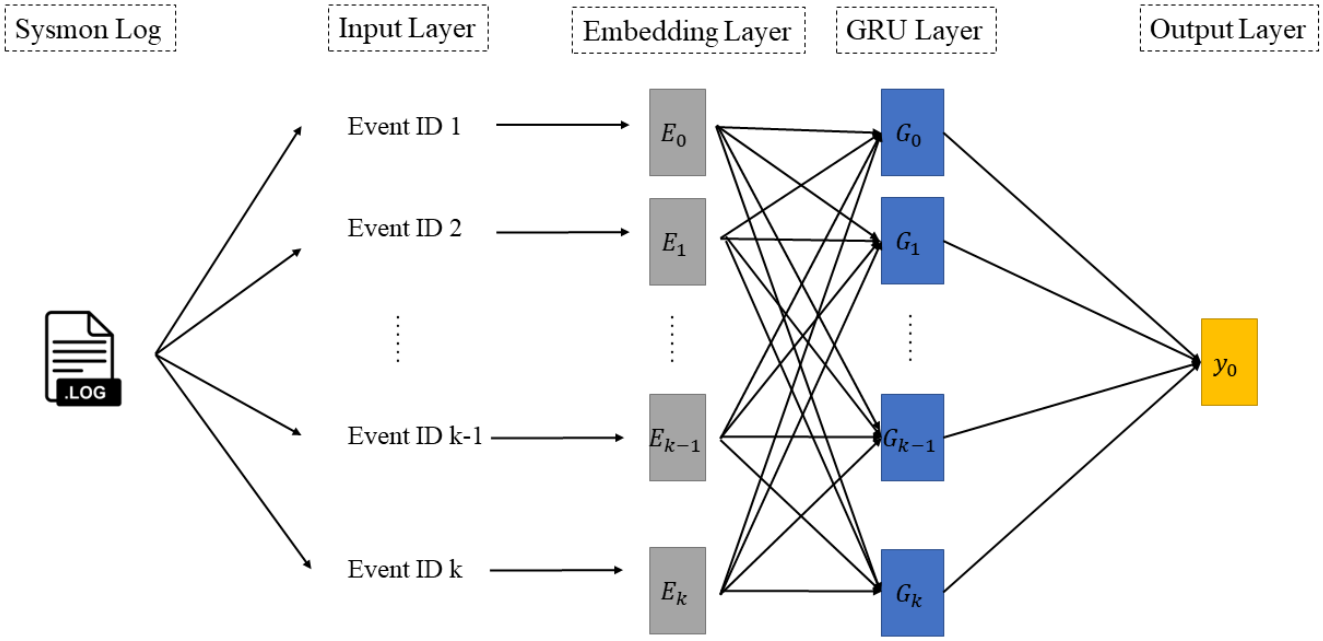


Figure 3: The proposed GRU model

Table 6: System parameters of the GRU model

Layer	Parameter	Parameter setting
Input layer	Input size	(None, ∞)
	Output size	(None,800)
Embedding layer	Input size	(None,800)
	Output size	(None,800,21)
	Mask zero	True
GRU layer	Input size	(None,800,21)
	Output size	(None,100)
	Dropout	0.2
Output layer	Input size	(None,100)
	Output size	(None,1)
	Activation	sigmoid
Other parameter setting	No. of Epochs: 60 Batch size: 400 Loss function: binary cross entropy Optimizer: adam Evaluation: accuracy	

Testing ranging from 2:8, 5:5, to 2:8) were tested and the detection results are plotted in Figure 4. The proposed model gives good accuracy even the training data is 20%, and its performance improves when it has more training data to learn anomalous behaviors.

$$AccuracyRate = \frac{TP + TN}{TN + FN + FP + TP} \quad (1)$$

4.2 Performance Comparison with SVM

The literature review informed that SVM performed well on anomaly detection and was selected as the baseline comparison in this study. The kernel function of SVM maps the data to a different space where a linear hyperplane can be used to separate classes. The RBF (radial basis function) is generally used kernel function and achieves a better performance comparing with other kernel functions. The parameters of SVM were optimized during the evaluation in order to build an SVM model with the best detection performance. The parameters of the best SVM model obtained are: the kernel function is Gaussian RBF, $\gamma = 0.00001$, and $C = 1000$, where the hyperparameter γ controls the tradeoff between error due to bias and variance in the SVM model and the hyperparameter C controls the trade-off between the slack variable penalty (misclassifications) and width of the margin. Each experiment was tested by random subsampling 5 times. The results of the performance comparison are outlined in Figure 5 and demonstrate that the proposed GRU model outperforms SVM and classifies both malicious and benign behaviors efficiently.

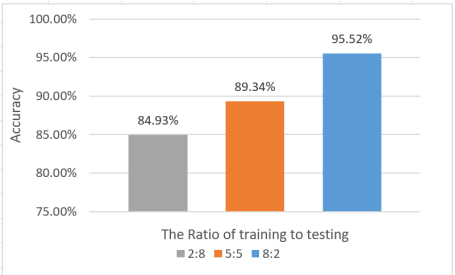


Figure 4: The detection performance on cross-validation

4.3 Performance Comparison with Human Analysis Report

To verify if the proposed system can identify valid malicious events, the generated results were compared

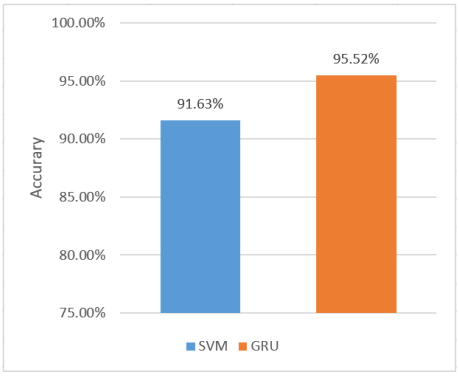


Figure 5: The performance comparison

with an analysis report [21] of PhotoMiner conducted by a security expert. Besides mining cryptocurrency, the malware PhotoMiner (screen.scr) exploited vulnerable FTP servers by password guessing attacks. The human analysis report indicates that the malware invoked cmd.exe to store the mining pool's information (pools.txt) to a temp folder and invoked NsCpuMiner32.exe mining and xcopy.exe spreading it to the disk device of the victim machine. Besides the above behaviors observed by the human analysis, the proposed system could provide additional detail information such as the locations of the suspicious programs and files as shown in Figure 6, where the malware screen.scr created NsCpuMiner32.exe and a html file and spawned a child process (cmd.exe) to create pools.txt at the temp folder.

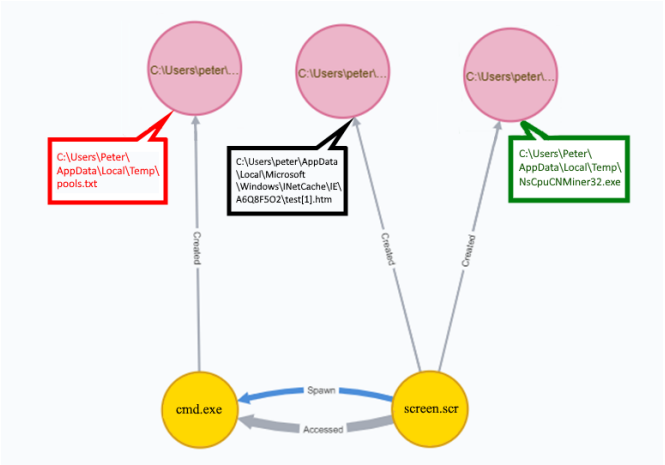


Figure 6: A portion of the detection results on file system

The detection results of the proposed system demonstrate that the malware spawned out many processes and provide a detailed parent-child process relationship. Figure 7 plots the spawned child processes in depth level one, where the red box denotes the malware process (screen.scr), the beige circle on its right indicates its location, the big green one concludes its child processes, each honey colored circle in the green box denotes a spawned child process, and the num-ber on a honey circle represents the locatioin of the child process. It can be seen that the malware spawned many xcopy.exe processes to spread the malware. The relationship of higher depth level can be produced as well.

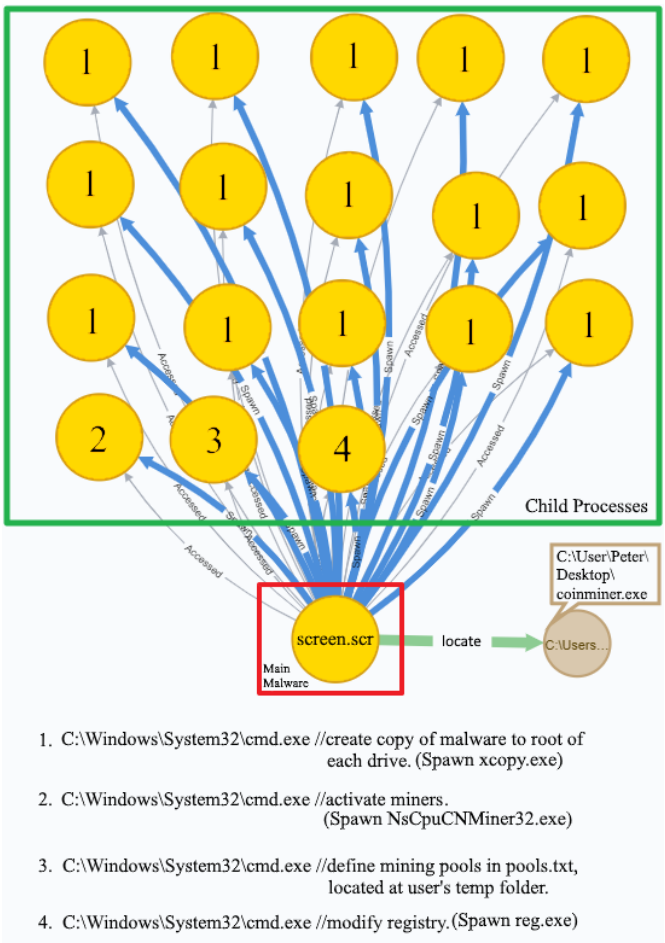


Figure 7: A portion of the detection results on process relationship

As for registry, the human analysis report identifies that the malware invoked cmd.exe executing reg.exe which created a new entry under HKCU\ SOFTWARE\Microsoft\Windows\CurrentVersion\Run in order to auto-start during system reboot. Besides this finding, the proposed system flagged more registry anomalies: network security setting, and multiple registry key values under HKU\ (User)\Software\Microsoft\Windows\CurrentVersion \Internet Settings\ ZoneMap, including ProxyBypass, IntranetName, UNCAsIntranet, AutoDetect, were modified. Because of the massive amount of registry keys, human experts could not identify all the distinguished key values without a valid tool, while the proposed system is useful and could identify registry changes. This comparison concludes that the proposed system could detect suspicious events efficiently.

Table 7: Detection results of unknown malware

Real / Detected	Malicious (Positive)	Benign (Negative)
malicious	97.15%	3.80%
benign	2.85%	96.20%
Accuracy	96.68%	

4.4 Evaluation of Unknown Malware Detection

1162 malware samples had not been analyzed and reported in VirusTotal and were considered as unknown malware. 1072 samples were able to run successfully on a sandbox environment, and a total of 1159 malicious log records was obtained. The same amount of benign behaviors were blended in order to evaluate if the proposed method can classify correctly on unknown malware as well as benign processes. Table 7 lists the detection results and demonstrates that the proposed detection model has a precision of 93.23% and a low false positive rate of 3.8%. The results prove that the proposed solution performs very well in detecting unknown malware.

5 Conclusion

In case of a security attack, it is time-critical matter to identify the cause and to reduce the impact of the damage. Audit logs are a reliable source to discover attacks and should be well-protected to prevent them from being compromised. Many organizations keep their important log files on cloud storage; hence data privacy becomes a concern. Efficient cryptography solutions, such as attribute-based encryption data sharing scheme based on Elliptic Curve Cryptography [5], are suitable for fulfilling the security requirement of cloud storage access.

Most organizations are lack of security experts and manpower to analyze a large number of audit trails and to discover suspicious events. This study proposes a GRU-based detection method that analyzes system logs and identifies malware misbehaviors.

The experiments emulated the attacks as well as normal usages in real-world environments. The experimental results indicate that the proposed solution classifies both benign and malicious behaviors efficiently, identifies unknown malware well, and outperforms SVM detection. In summary, the performance evaluation demonstrates that the proposed machine learning model is practical and efficient. Future research could enhance log analysis and anomaly detection by including additional log files and expertise knowledge to improve detection performance.

References

[1] H. H. T. Albasheer, M. M. Siraj and M. M. Din, "Towards predictive real-time multi-sensors intrusion alert correlation framework," *Indian Journal of Science and Technology*, vol. 8, no. 12, 2015.

[2] M. Amini, J. Rezaeenoor, and E. Hadavandi, "Effective intrusion detection with a neural network ensemble using fuzzy clustering and stacking combination method," *Journal of Computing Security*, vol. 1, no. 4, pp. 293-305, 2014.

[3] S. D. D. Anton, S. Sinha, and H. D. Schotten, "Anomaly-based intrusion detection in industrial data with svm and random forests," *Cryptography and Security*, 2019. (`https://arxiv.org/abs/1907.10374`)

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *Computation and Language*, 2014. (`https://arxiv.org/abs/1406.1078`)

[5] M. A. Doostari, S. Rezaei and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, pp. 115–131, 2018.

[6] J. Dwyer and T. M. Truta, "Finding anomalies in windows event logs using standard deviation," in *The 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 563–570, 2013.

[7] Infosec Institute, *Common Malware Persistence Mechanisms*, Technical report, 2016. (`https://resources.infosecinstitute.com/common-malware-persistence-mechanisms/#gref`)

[8] T. Y. Kim and S. B. Cho, "Web traffic anomaly detection using C-LSTM neural networks," *Expert Systems with Applications*, vol. 106, pp. 66–76, 2018.

[9] C. C. Lo, C. C. Huang, and J. Ku, "A cooperative intrusion detection system framework for cloud computing networks," in *The 39th International Conference on Parallel Processing Workshops*, pp. 280–284, 2010.

[10] Microsoft, *Structure of the Registry*, Technical report, 2018. (`https://docs.microsoft.com/en-us/windows/win32/sysinfo/structure-of-the-registry`)

[11] N. Mishra A. Tayal and S. Sharma, "Active monitoring & postmortem forensic analysis of network threats: A survey," *International Journal of Electronics and Information Engineering*, vol. 6, pp. 49–59, 2017.

[12] F. Nabi and M. M. Nabi, "A process of security assurance properties unification for application logic," *International Journal of Electronics and Information Engineering*, no. 6, pp. 40–48, 2017.

[13] V. Nigam, *Understanding Neural Networks. From Neuron to RNN, CNN, and deep learning*, Technical report, 2018. (`https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90`)

[14] P. Prasse, L. Machlica, T. Pevný, J. Havelka, and T. Scheffer, "Malware detection by analysing network traffic with neural networks," in *IEEE Security and Privacy Workshops (SPW'17)*, pp. 205–210, 2017.

[15] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, "Network traffic anomaly detection using recurrent neural networks," *Computers and Society*, 2018. (https://arxiv.org/abs/1803.10769)

[16] E. Raftopoulos and X. Dimitropoulos, "IDS alert correlation in the wild with EDGe," *IEEE Journal on Seleted Areas in Communi-cations*, vol. 32, no. 10, pp. 1933-1946, 2014.

[17] N. Serketzis, V. Katos, C. Ilioudis, D. Baltatzis, and G. Pangalos, "Towards a threat intellegence informed digital forensics readiness framework," in *Twenty-Fifth European Conference on Information Systems (ECIS'17)*, 2017. (http://eprints.bournemouth.ac.uk/30391/)

[18] A. Singhal, C. Liu and D. Wijesekera, "A model towards using evidence from security events for network attack analysis," *International Workshop on Security in Information System*, pp. 83–95, 2014. (https://doi.org/10.5220/0004980300830095)

[19] Statcounter, *Desktop Operating System Market Share Worldwide*, Technical report, 2020. (http://gs.statcounter.com/os-market-share/desktop/worldwide)

[20] Thycotic, *Black Hat 2018 Hacker Survey Report*, Technical report, 2018. (https://go.thycotic.com/l/101722/2018-09-12/5gf8wq/101722/74015/Report_2018_Black_Hat_Survey.pdf?_ga=2.52829416.1772536159.1560412381-274843393.1560412381)

[21] Taiwan Academic Network Computer Emergency Response Team, *Incident Analysis Report of Miner Trojan Photominer Infecting Campus Machines*, Technical report, 2017. (https://portal.cert.tanet.edu.tw/docs/pdf/201709290109555585771228906051.pdf)

[22] H. Wang, J. Gu, and S. S. Wang, "An effective intrusion detection framework based on svm with feature augmentation," *Knowledge-Based Systems*, vol. 136, pp. 130–139, 2017.

[23] S. T. Zargar, H. Takabi, and J. B. D. Joshi. "DCDIDP: A distributed, collaborative, and data-driven intrusion detection and prevention framework for cloud computing environments," in *The 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'11)*, pp. 332–341, 2011.

[24] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through SVM classifier," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 340–347, 2017.

Biography

Chia-Mei Chen has joined in the Department of Information Management, National Sun Yat-Sen University since 1996. She was a Section Chef of Network Division and Deputy Director, Office of Library and Information Services in 2009-2011. She had served as a coordinator of TWCERT/CC (Taiwan Computer Emergency Response Team/Coordination Center) during 1998 to 2013 and established TACERT (Taiwan Academic Network Computer Emergency Response Team) in 2009. She had served as the Deputy Chair of TWISC@NCKU, a branch of Taiwan Information Security Center, for three years. She continues working for the network security society. Her current research interests include anomaly detection, malware analysis, network security, and cyber threat intelligence.

Gen-Hong Syu was a graduate student at the Department of Information Management, National Sun Yat-sen University. He is interested in digital forensics.

Zheng-Xun Cai received master degree at National Sun Yat-sen University, Kaohsiung, Taiwan in 2017. Now he is PhD student in National Sun Yat-sen University, Kaohsiung, Taiwan. His research interests involve digital forensics, network analysis and process analysis.

Efficient and Secure Outsourcing of Modular Exponentiation Based on Smart Contract

Danting Xu, Yanli Ren, Xiangyu Li, and Guorui Feng
(Corresponding author: Yanli Ren)

School of Communication and Information Engineering, Shanghai University,
No. 99, Shangda Road, Baoshan District, Shanghai 200444, China
(Email: renyanli@shu.edu.cn)

Abstract

Securely and effectively outsourcing Modular exponential (MExp) computation which is one of the most complex operations in public key cryptography to an untrusted server is popular in recent years. Based on smart contract in the blockchain, we propose an efficient outsourcing algorithm of MExp with single server. With the mechanism of smart contract, the results returned by blockchain after the smart contract execution must be right. There is no need for verifying the results since the checkability has been 1, which greatly reduces the computational cost of outsourcer and server. We give a strict security proof of the scheme and make the implement on Hyperledger Fabric. Both theoretical analysis and experiment results indicate that our algorithm improves the computation efficiency of outsourcer and server, protects the secrecy of input and output data, and keeps the outsourcing results valid with a probability of 1.

Keywords: Hyperledger Fabric; Modular Exponentiation; Secure Outsourcing Computation; Smart Contract

1 Introduction

Accompanied by the rapid development of the technology about digital information, computation power becomes an essential resource today. Almost all digital information engineering involves important and complex computation, however, it is often difficult for lightweight mobile devices to afford [14]. At this point, the clients with limited resource are willing to be an outsourcer and outsource computation they can't afford to a powerful server by paying a certain fee. The processing of computation outsourcing realizes a more reasonable allocation of resources, saves a lot of time for outsourcer and benefits the server. Especially, with the development of cloud computing, outsourcing computation to cloud server with powerful computing resource has become an important computing mode [8, 23, 28].

However, excepting solving complex computing for outsourcer and improving their computing efficiency, outsourcing computation also brings the challenge of data security protection. The challenges come mainly from the secrecy of the data and the checkability of computation results [7]. Firstly, the server is untrusted to the outsourcer. Once the computing process involves some private information, the privacy of outsourcer may be leaked [11]. Therefore, a secure outsourcing algorithm must ensure the input and output data cannot be directly obtained by the server to protect the privacy of outsourcer. Secondly, server may return invalid computation results for saving workload or colluding with a third party. So, outsourcer needs to check if the computation results from the server are valid [9]. Meanwhile, in the whole outsourcing computing processing, the computational complexity of outsourcer must be much lower than that of non-outsourcing computing, making sure the operability and significance of outsourcing computation [19]. Therefore, how to design an outsourcing scheme covering the data privacy, the results checkability and the operability has aroused great attention of academic and industrial circles. There is a broad prospect about research of secure outsourcing computing.

1.1 Related Work

In the field of cryptography, there are many extremely complex computations, one of which is modular exponential computation. Using the method of square-and-multiply, single modular exponentiation (MExp) will take approximately 1.5L modular multiplications (MMs), in which L represents the bit length of the exponent [16]. Secure outsourcing of modular exponentiation is a focal point in the research of outsourcing computation. There are mainly two directions in the research field, which are secure outsourcing of MExp based on multiple servers and based on single server [29]. Especially in recent years, secure outsourcing algorithms of MExp based on single server develops rapidly. Dijk *et al.* [10] proposed an outsourcing algorithm for MExp based on single server, which was however unable to protect the secrecy of inputs information. Then, Wang *et al.* [24] gave a scheme for outsourcing MExp based on single server. Though it pro-

tected the secrecy of inputs, its efficiency was low and checkability was just 1/2. At that time, researchers hoped to improve the efficiency and checkability of MExp outsourcing algorithm with single server without losing the secrecy of inputs and outputs. Kiraz *et al.* [15] presented more than one algorithm of outsourcing MExp based on single server with an improved checkability which was still lower than 1. Cai *et al.* described an algorithm in [5] for outsourcing MExp based on single server, but its efficiency was still low and there would be many queries from outsourcer to the server. Ye *et al.* proposed two algorithms of outsourcing MExp with two servers and single server, respectively [26, 27]. The checkability was improved to a large extent in their algorithms, however, the outsourcer needed to pay a large cost of computation. Ren *et al.* [21]. presented two more efficient outsourcing algorithms of MExps with a checkability of 1 based on two servers and single server respectively. However, the outsourcer still has to make much computation to check if the results from sever are valid, which make the algorithm complex.

Though the checkability of MExp outsourcing scheme based on single server which presented by Ren *et al.* is the highest, it causes a huge computational cost while keeps data secret to server and computation results verified by outsourcer. Obviously, if the computation results must be right without any verification, the computational cost will be greatly reduced. Smart contract, an automatically executing program according to contract contents gives sprint to us. The inherent properties of smart contract guarantee that the executing results from it must be correct and there is no need for any correctness check of executing results.

In recent years, blockchain technology develops rapidly. Unlike traditional centralized systems, there will be a public distributed ledger maintained by the entire network based on consensus mechanism to record the state of blockchain system. The validity of any system behavior needs authentication from all nodes in the system network. Therefore, it can maintain correct system state without any trust mechanism [6]. In particular, smart contract as the important features in second generation of blockchain breaks the restriction in the first generation of blockchain, which is represented by bitcoin and only applicable to specific application scenarios finance [12]. Specifically, it is the Ethereum, which is the representative of second generation of blockchain, provides a platform for developers to build blockchain applications in line with their own needs by programming smart contract [2]. Smart contract is proposed by Nick [18] in 1995 firstly. It is essentially a computer program which can execute according to the trigger conditions those have been programmed on it automatically. It is deployed to blockchain, and after invoked, it can be executed by all nodes in blockchain without any central mechanism. Ideally, smart contracts work in strict accordance with what programs do, so the executing results from it must be correct. Of course, Ethereum is a powerful platform to program smart contract and construct blockchain application. But the programming language is solidity on it, which is not very friendly to developers. Hyperledger Fabric, another popular blockchain programming platform, provides programming languages of Java, Go and other traditional languages, which is more programming and applicating friendly [3].

Our contributions. In this paper, based on smart contract, we present a secure outsourcing algorithm for modular exponentiation whose checkability is 1. We choose smart contract to replace the outsourcing server. Different from the traditional untrusted server, smart contract is totally trusted to outsourcer. When the outsourcer outsources modular exponentiation to smart contract, they needn't verify the validity of the computation results, which will greatly reduce the complexity of the algorithm. Compared with previous algorithms, the scheme we propose has the higher efficiency while keeping the checkability of 1. Despite the theoretical algorithm construction, we make a simulation experiment of our scheme on Hyperledger Fabric, a popular blockchain platform supporting the programming and running of smart contract. We give the detail data about the checkability and computing efficiency of our scheme and make comparison with previous algorithms of checkability and computing efficiency to show the advantages of our scheme.

This paper is organized as follows. In Section 2, we give some definitions of secure outsourcing computation and introduce smart contract. The proposed algorithm and comparisons are given in Section 3. Then we do some experiments to evaluate the performance of proposed algorithms in Section 4. Finally, conclusion is made in Section 5.

2 Security Model and Smart Contact

In this section, we give the details about the definitions of secure outsourcing computation, the secure outsourcing computation model, the Euler theorem which is the important theoretical basis of our algorithms and the smart contract.

2.1 Definition and Security Model of Secure Outsourcing Computation

The formal definitions of secure outsourcing computation were given by Hohenberger and Lysyanskaya [13] firstly, which will be used in our algorithm.

Firstly, we name the algorithm Alg, two parties in Alg T and U.T is the outsourcer which is trusted but has limited computational ability. U is an untrusted server with rich computational resource. T can invoke U , denoted as T^U. T and U will make an implement of Alg, however, there will be a server named U'. U' tries to replace U, working maliciously and recording all computations when it is invoked. (T, U) is regarded as a se-

cure outsourcing implementation of Alg when U' could get nothing about the inputs and outputs of $T^{U'}$ during computing.

Then we will give the specific definition of secure computing outsourcing which was proposed by Hohenberger and Lysyanskaya in [13].

Definition 1. *(Algorithm with Outsource-IO). An algorithm which gets 5 inputs and generates 3 outputs will be called complying the specification of outsourcing input/output. Meanwhile, according to the degree of known to adversary $A = (E, U')$, the inputs and outputs are distinguished. Thereinto, E represents the adversarial environment.*

The specific classification of the inputs and outputs is as the following introduction. At first, there are three inputs created by an honest party. The one protected from not only E but also U' is regarded as an honest, secret input; the one may be unprotected from E, but protected from U' is regarded as an honest, protected input; The one may be unprotected from both E and U' is regarded as an honest, unprotected input. Meanwhile, E also generates two inputs. The one unprotected from E, but protected from U' is regarded as an adversarial, protected input; the other unprotected from both E and U' is regarded as an adversarial, unprotected input.

Analogously, the output protected from E and U' is regarded as secret; the output may be unprotected from E, but not U' is protected is regarded as protected; And the output may be unprotected from both E and U' is regarded as unprotected.

Definition 2. *(Outsource-security). If (T, U) which is a pair of algorithms with outsourcing I/O satisfies the following conditions, we say it is an outsource-secure implementation of Alg.*

- *Correctness: T^U is a correct implementation of Alg.*
- *Security: For any adversary $= (E, U')$, there are simulators (S_1, S_2) which can make it computationally indistinguishable for the following pairs of random variables.*
  Pair One : $EVIEW_{real} \sim EVIEW_{ideal}$

The adversarial environment E is unable to get any information about input or output when it executes T^U. Real process and ideal process go ahead in turn. The real process:

$$\begin{aligned} EVIEW^i_{real} = \{&(istate^i, x^i_{hs}, x^i_{hp}, x^i_{hu}) \\ &\leftarrow I(1^k, istate^{i-1}); \\ &\quad (estate^i, j^i, x^i_{ap}, a^i_{au}, stop^i) \\ &\leftarrow E(1^k, EVIEW^{i-1}_{real}, x^i_{hp}, x^i_{hu}); \\ &\quad (tstate^i, ustate^i, y^i_s, y^i_p, y^i_u) \\ &\leftarrow T^{U'(ustate^{i-1})} \times (tstate^{i-1}, x^i_{hs}, x^i_{hp}, x^i_{hu}, x^i_{ap}, x^i_{au}) : \\ &\quad (estate^i, y^i_p, y^i_u)\} \end{aligned} \tag{1}$$

$EVIEW_{real} = EVIEW^i_{real}$ if $stop^i = TRUE$;

At round i in the real process, the honest, stateful process I chooses the "honest, secret", "honest, protected", and "honest, unprotected" inputs $(x^i_{hs}, x^i_{hp}, x^i_{hu})$ which couldn't be obtained by the environment E. Then, based on $EVIEW^i_{real}$, the previous round of state, and the inputs $(x^i_{hs}, x^i_{hp}, x^i_{hu})$ of $T^{U'}$, E can choose $estate^i$ (a variable to remind the next operation), j^i (an index), x^i_{ap}, x^i_{au} (two inputs of adversary A), $stop^i$ (a Boole function that declares if round i is the last one). After that, on the inputs $(tstate^{i-1}, x^i_{hs}, x^i_{hp}, x^i_{hu}, x^i_{ap}, x^i_{au})$, the algorithm $T^{U'}$ produces a new state $tstate^i$, the secret y^i_s, protected y^i_p and unprotected y^i_u outputs, where $tstate^{i-1}$ is the previous state of T. Based on the previous state $ustate^{i-1}$, U' saves the current state as $ustate^i$. The output at round i is $(estate^i, y^i_p, y^i_u)$ for the real process , and in the whole real process, the output of U is the output of the last round, so $stop^i = TREU$. The ideal process:

$$\begin{aligned} EVIEW^i_{ideal} = \{&(istate^i, x^i_{hs}, x^i_{hp}, x^i_{hu}) \leftarrow I(1^k, istate^{i-1}); \\ &\quad (estate^i, j^i, x^i_{ap}, a^i_{au}, stop^i) \\ &\leftarrow E(1^k, EVIEW^{i-1}_{ideal}, x^i_{hp}, x^i_{hu}); (astate^i, y^i_s, y^i_p, y^i_u) \\ &\leftarrow Alg(astate^{i-1}, x^i_{hs}, x^i_{hp}, x^i_{hu}, x^i_{ap}, x^i_{au}); \\ &\quad (sstate^i, ustate^i, Y^i_p, Y^i_u, replace^i) : \\ &\leftarrow S_1^{U'(ustate^{i-1})} \times (sstate^{i-1}, x^i_{hp}, x^i_{hu}, x^i_{ap}, x^i_{au}, y^i_p, y^i_u); \\ &\quad (z^i_p, z^i_u) = replace^i(Y^i_p, Y^i_u) + (1 - replace^i)(y^i_p, y^i_u) : \\ &\quad (estate^i, z^i_p, z^i_u)\} \end{aligned} \tag{2}$$

$EVIEW_{ideal} = EVIEW^i_{ideal}$ if $stop^i = TRUE$.

In ideal process, simulator S_1 cannot get secret input x^i_{hs}, but it can get the protected and unprotected output of algorithm Alg at round i. Then S_1 can choose to output (y^i_p, y^i_u) or replaces it with another value (Y^i_p, Y^i_u) according to Boole variable $replace^i$. During the whole process, U' can be invoked by S_1 while whose state may be stored by U' as the real process.

Pair Two : $UVIEW_{real} \sim UVIEW_{ideal}$

Untrusted software U' programmed by the adversarial environment E is unable to get any information about inputs or outputs when it executes T^U.

Similar to pair one, if $stop^i = TREU$, U' in the real process just has the state $UVIEW_{real} \sim ustate^i$. The ideal process:

$$\begin{aligned} UVIEW^i_{ideal} = \{&(istate^i, x^i_{hs}, x^i_{hp}, x^i_{hu}) \\ &\leftarrow I(1^k, istate^{i-1}); (estate^i, j^i, x^i_{ap}, a^i_{au}, stop^i) \\ &\leftarrow E(1^k, estate^{i-1}, x^i_{hp}, x^i_{hu}, y^{i-1}_p, y^{i-1}_u); \\ &\quad (astate^i, y^i_s, y^i_p, y^i_u) \\ &\leftarrow Alg(astate^{i-1}, x^i_{hs}, x^i_{hp}, x^i_{hu}, x^i_{ap}, x^i_{au}); \\ &\quad (sstate^i, ustate^i) \\ &\leftarrow S_1^{U'(ustate^{i-1})} \times (sstate^{i-1}, x^{j^i}_{hu}, x^i_{au}) : (ustate^i)\} \end{aligned} \tag{3}$$

$UVIEW_{ideal} = UVIEW^i_{ideal}$ if $stop^i = TRUE$.

In the ideal process, simulator S_2 can only obtain the unprotected inputs (x^i_{hu}, x^i_{au}) and it has right to make queries to U'. Similar to $PairOne$, it is possible that U' keeps its state.

Definition 3. *(α-Efficient, Secure Outsourcing). Algorithm (T, U) is an α-efficient implementation of Alg when T^U is a correct implementation of Alg and for any input x, multiplicative factor of T's running time than $Alg's'$ is no more than α.*

Definition 4. *(β-Checkable, Secure Outsourcing). Algorithm (T, U) is a β-checkable implementation of Alg when it is a correct implementation of Alg, and for any input x, the probability for T could catch any error once U' work maliciously is no less than β.*

Definition 5. *((α, β)-Outsource-Security). Algorithm (T, U) is an (α, β)-outsource-secure implementation of Alg when it is not only α-efficient but also β-checkable.*

Analogous to the *two untrusted program model* presented by Hohenberger *et al.* [13], in our algorithm based on single server, E programs an application as the untrusted servers U' and sends it to T, then T installs it in a manner. The adversary is $A = (E, U')$ where U' is the untrusted server. Our algorithm is secure in this model.

2.2 Euler Theorem

Euler Theorem [22] is a theorem about identity, which was named after the Swiss mathematician Leonhard Euler. It is considered as one of the most beautiful theorems in the mathematical world. Euler Theorem is actually an extension of Fermat's little theorem.

Euler Theorem is the important computing features in the proposed algorithm. We will make a brief introduction of it here, please refer to [22] for details.

Theorem 1. *Supposing x and y are two positive integers, and x is co-prime with y, we can get that $x^{\varphi(y)} \equiv 1(mody)$, thereinto, $\varphi(y)$ denotes the number of integers which are smaller than y and co-prime with y. The function φ is known as the Euler function of y [20].*

Euler Theorem can greatly simplify some computing operations, for example, solving the multiplication inverse, reducing the power when modulo and so on. Therefore, it is widely applied in cryptography, economics and so on. It is also the core of the famous public key encryption algorithm named RSA. The correctness of encryption and decryption for RSA is just based on *Euler Theorem*.

2.3 Smart Contract

Smart contract was firstly presented by Nick [18] in 1995. Then, it becomes the core of the second generation of blockchain. Essentially, on the one hand, it is a contract including a set of commitments defined digitally from contract participants. On the other hand, it is a program which can be executed automatically once it meets the trigger condition. In another word, ideally, no matter whatever clients want to do, they can make a contract according to their specific requirement. Once they reach an agreement of the contract, the contract can be programmed to a smart contract, which will automatically execute strictly according to its content [12].

But how does smart contract work on blockchain? Firstly, developers who construct the blockchain application will program and code the smart contract on the blockchain programming platform just like Ethereum or Hyperledger Fabric according to the specific contract contents which participants have reached an agreement on. Secondly, the smart contract will be deployed to blockchain. Just as a transaction in blockchain network, the smart contract will be ultimately packed to chain through the mining by blockchain nodes. Thirdly, clients can call the specific smart contract they want by the unique account address of it. The operation is also built as a transaction, including the account address of smart contract clients want to call, the function and input clients hope the contract to execute, the account address of clients and the cost clients will pay for the transaction. Finally, when the transaction of invoking smart contract is packed to the chain, the smart contract will be executed by all blockchain network participant nodes. Achieving the executing, all nodes in blockchain will update the data state.

The information about deploying and invoking of smart contract is all packed as a transaction to blockchain, which is just like the Phase 1 and Phase 2 show in Figure 1. According to the consensus mechanism of blockchain, all the transaction will be put on the chain and recognized to be valid only when all network nodes have verified it and reach a consensus. In another word, the deploying and invoking of smart contract are all certificated by the whole blockchain system and they must be legal. After being deployed and invoked, smart contract will be executed automatically by all the nodes in blockchain network. The results of smart contract from all the nodes must coincide to each other or they will be invalid in the blockchain system.

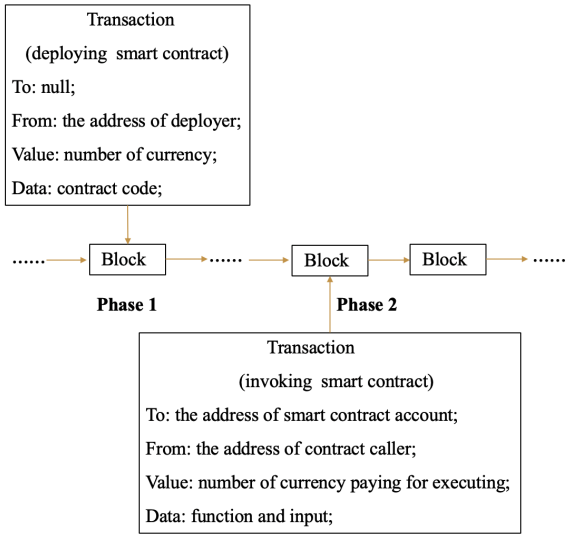


Figure 1: Deploying and invoking Smart contract on the blockchain

Even if there are some incorrect results from malicious nodes, they will not be recognized.

In conclusion, even if there is no trust between smart contract and the clients who call it, the executing results from smart contract are trusted.

3 Efficient Outsourcing of Modular Exponentiation based on Smart Contract

In this section, we firstly construct a specific outsourcing model which uses smart contract as the single outsourcing server. Then we present a secure outsourcing algorithm of modular exponentiation based on smart contract with a checkability of 1. Finally, we give security analysis for the proposed algorithm and compare its efficiency and checkability with those of the previous ones.

3.1 An Outsourcing Model with Smart Contract

In our outsourcing model, we choose smart contract which has implemented modular exponentiation computation to replace the untrusted server the previous algorithm used to give the computation results of specific modular exponentiation.

In the outsourcing model with smart contract shown in Figure 2, the Blockchain System with smart contract implementing MExp supports the deploying and executing of smart contract, which is specifically the blockchain application programming platform just like Ethereum or Hyperledger Fabric and so on, and based on the features of blockchain, it is trusted to outsourcer. The Outsourcer is the party who outsources the modular exponentiation computation he can't afford.

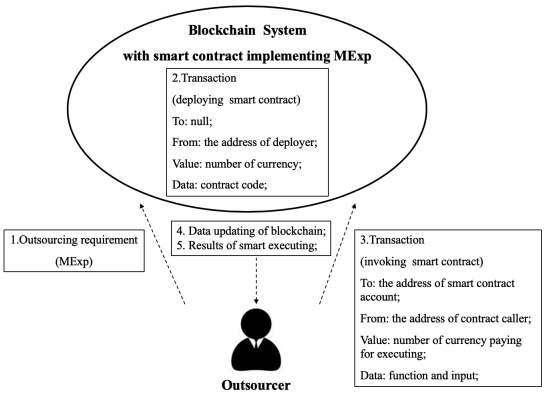


Figure 2: An outsourcing model with smart contract

An outsourcing model mainly includes the following steps:

1) The Outsourcer sends outsourcing requirement to the Blockchain System with smart contract implementing MExp;

2) Receiving the computation requirement, nodes in Blockchain System could make a transaction of deploying the smart contract. Once the transaction is packed to chain, the smart contract will be recognized by the whole blockchain network and it can be invoked and executed by all of the nodes in the blockchain;

3) After being deployed, as a client in the blockchain, the outsourcer can make a transaction to invoke the specific smart contract by the account address of it to execute the computation.

4) After receiving the transaction of invoking the smart contract on the chain, the nodes in the blockchain will execute it. When the results reach an agreement under the consensus mechanism by all of the nodes in the network, they will recognize the results and update the data status of the blockchain.

5) Finally, the results will be sent to the outsourcer who invokes the smart contract as an output of the smart contract.

As introduced in Subsection 2.3, the results from smart contract are trusted to the outsourcer, and the validity verification of the outsourcing results is not needed. Besides protecting the inputs and outputs, there is no need for the outsourcing algorithm to make sure whether the computation results are valid. Therefore, the algorithm of outsourcing MExp based on smart contract obviously has the lower complexity and the higher efficiency while keeping the outsourcing inputs and outputs secret.

3.2 The Proposed Outsourcing Algorithm

Then we present the specific outsourcing algorithm of MExp based on smart contract with a checkability of 1.

In our algorithm, there are two large primes p, q and $q|p-1$. Similar to the subprogram named Rand in [9], we have a subprogram named $Rand'$ to make the generation of a random tuple as follows:

$$\{(\alpha, g^{\alpha}, g^{-\alpha}\}, (\beta, g^{\beta})(t_1^{-1}, g^{t_1}), (\xi_j, g^{-\xi_j}), \mu_j, g^{\sum_{i\in A}\xi_i\mu_i}\}$$

where $j = \{1, 2, \ldots, b\}$, $A \subset \{1, 2, \ldots, b\}$, and b is a positive, $\alpha, \beta \in Z_q$, g is a generator of Z_p^*. The specific realization of the subroutine can utilize EBPV generator [17] or the table-lookup method [1].

In our presented algorithm, the inputs are $a \in Z_q$ and the output is $u^a mod p$, where $u \in Z_p^*$ and $u^q \equiv 1(mod\ p)$. To protect the privacy of the outsourcer, u and a are all private to the smart contract. When receiving the inputs (x, y), it outputs $y^x mod\ p$. Specifically, the outsourcer does the following operations to outsource modular exponentiation:

1) T invokes $Rand'$ to create a tuple

$$\{(\alpha, g^{\alpha}, g^{-\alpha}\}, (\beta, g^{\beta})(t_1^{-1}, g^{t_1}), (\xi_j, g^{-\xi_j}), \mu_j, g^{\sum_{i\in A}\xi_i\mu_i}\}$$

where $j = \{1, 2, \ldots, b\}$, $A \subset \{1, 2, \ldots, b\}$, and gets the first logical division:

$$u^a = (vw)^a = g^{\alpha a} w^a = g^{\beta} g^{\gamma} w^a \tag{4}$$

where $v = g^{\alpha}, w = uv^{-1}, \gamma = \alpha a - \beta$.

2) To make the exponent a blind to smart contract, T chooses a set of $\mu_i, i \in A$, which is from the set of random numbers $\mu_j, j = \{1, 2, \ldots, b\}$ according to the $Rand'$ subroutine. Then T computes

$$r_1 = a - \sum_{i \in A} \mu_u$$

Therefore, another logical division is:

$$u^a = g^{\beta} g^{\gamma} w^a = g^{\beta} g^{\gamma} w^{r_1} + \sum_{i \in A} \mu_i (mod\ p)$$

3) Next, T computes

$$w_j = w g^{-\xi_j}, j = \{1, 2, \ldots, b\}$$

4) Then T makes invocations of U in random order, where U is the smart contract which has been programmed to compute modular exponentiation:

$$\begin{aligned} U(\gamma t_1^{-1}, g^{t_1}) &\rightarrow \eta_1 = g^{\gamma} (mod\ p) \\ U(r_1, w) &\rightarrow \eta_2 = w^{r_1} (mod\ p) \\ U(\mu_j, w_j) &\rightarrow w_j^{\mu_j} (mod\ p), j = \{1, 2, \ldots, b\} \end{aligned}$$

As introduced in 2.3, the results from smart contract are trusted for the outsourcer. Therefore, T can multiply the outputs of U without the verification process to get the result of $u^a mod\ p$ as follows:

$$\begin{aligned} & g^{\beta} \cdot \eta_1 \cdot \eta_2 \cdot (\Pi_{i \in A} w_i^{\mu_i}) \cdot g^{\sum_{i \in A} \xi_i \mu_i} \\ = \ & g^{\beta} \cdot g^{\gamma} \cdot w^{r_1} \cdot \prod_{i \in A} \left(w g^{-\xi_i}\right)^{\mu_i} \cdot g^{\sum_{i \in A} \xi_i \mu_i} \\ = \ & g^{\beta} \cdot g^{\gamma} \cdot w^{r_1 + \sum_{i \in A} \mu_i} = u^a (\mathrm{mod} p). \end{aligned}$$

Remark 1. *As introduced in 3.2, we can see that security of the outsourcing algorithm has a tight relation to the value of b. As shown in [24], the bit length of a random number must be no less than 64. In another word, the possible values of a random number should be more than* 2^{64}*. To meet this requirement, the parameter b should be at least 53 in our algorithm based on smart contract. Now we give the details of the computing the value of b.*

As shown in 3.2, $a \equiv r_1 + \sum_{i \in A} \mu_i (\mathrm{mod} q)$, which means that security of the presented algorithm is decided by the value of r_1 and $\sum_{i \in A} \mu_i$.

For the smart contract, since the $b+2$ queries from T is executed in random order, the number of possible values for r_1 is $b+2$. Having the value of r_1, the smart contract should try to the find all of $\mu_i (i \in A)$ from the set of $\mu_j (j = 1, 2, \ldots, b)$ and compute $\sum_{i \in A} \mu_i$. So the number of possible values for $\sum_{i \in A} \mu_i$ is $C_{b+1}^b \cdot \sum_{k=1}^b \cdot C_b^k$, where C_{b+1}^b denotes the possible values of selecting b queries from $b+1$ queries and C_b^k denotes possible values of picking k integers from b integers. In conclusion, there are $(b+2) \cdot C_{b+1}^b \cdot \sum_{k=1}^b \cdot C_b^k$ possible number of a for the smart contract, and it approximately equals 2^{64} when $b = 53$. Therefore, we set b should be at least 53 to make sure the security of the algorithm we proposed.

3.3 Security Analysis

Theorem 2. *In the outsourcing algorithm based on single server (smart contract), no matter the input* (a, u) *is "honest, secret", "honest, protected" or "adversarial, protected",* (T, U) *is an outsource-secure implementation of the presented algorithm.*

Proof. As same as [24], $A = (E, U')$ is an adversary who makes interaction with the outsourcing algorithm in the security model based on single server. □

Pair One: $EVIEW_{real} \sim EVIEW_{ideal}$.

$EVIEW_{real} \sim EVIEW_{ideal}$ means that environment E can't learn anything from the execution of (T, U). If the input (a, u) is "honest, protected" or "adversarial, protected", the behavior of the simulator S_1 is same as that executed in the real execution. In another word, what we only need to prove is that it still holds when (a, u) is "honest, secret".

Assuming the input (a, u) is honest and secret, the behavior of simulator S_1 is shown as follows. After obtaining the inputs of round i, S_1 ignores them and invokes U randomly for $b+2$ times instead. Then the simulator S_1 tests all of the outputs U' returns. If S_1 detects an error, the simulator S_1 stores the states and outputs of it, $Y_p^i =$ "$error''$", $Y_u^i = \emptyset$, $replace^i = 1$. If there is no error detected, S_1 outputs $Y_p^i = \emptyset, Y_u^i = \emptyset, replace^i = 0$; otherwise, S_1 picks a random element $r \in Z_p^*$, gives the output $Y_p^i = r, Y_u^i = \emptyset$ and makes $replace^i = 1$. In this case, S_1 also stores its states. In the ideal experiment, the inputs are picked by T randomly; in the real one, all operations of invoking the server made by T are re-randomized, in another word, they are all computationally random. So, in fact, the input of U' in both the ideal and the real experiments are all computationally indistinguishable. At round i, if U' is honest in the real experiment, $T^{U'}$ executes the algorithm correctly and S_1 won't replace the outputs, and there is no doubt that $EVIEW_{real} \sim EVIEW_{ideal}$. If U' isn't honest at round i, all failures are known for T (from blockchain) and S_1, and the outputs will just be "error". In the real experiment, along with a random value r, all outputs amounting to $|A| + 2$ returned by U' will be multiplied together. Though U' behaves dishonestly, $EVIEW_{real} \sim EVIEW_{ideal}$ still holds. With the hybrid argument, we believe that $EVIEW_{real} \sim EVIEW_{ideal}$.

Pair Two: $EVIEW_{real} \sim EVIEW_{ideal}$.

The simulator S_2 behaves as follows. After obtaining the inputs at round i, S_2 ignores them and invokes U' randomly for $b+2$ times instead. Then S_2 stores the states of its own and U'. E can easily find any difference between the ideal experiments and the real ones, however, there can't be any interaction between E and U. So, at each round, we have $EVIEW_{real} \sim EVIEW_{ideal}$.

Theorem 3. *(T,U) is an $(O(\frac{\log^2 m}{m}),1)$-outsource-secure implementation of the algorithm for outsourcing modular exponentiation computation based on smart contract.*

Proof. The presented algorithm based on smart contract has one call to $Rand'$ and $b+|A|+5$ modular multiplications (MMs) to compute $u^a mod p$. There are $O(\log^2 m)$ MMs during the invocation of $Rand'$ if we choose the EBPV generator while there are $O(1)$ MMs if we choose the method of table-lookup. So (T,U) is an $O(\frac{\log^2 m}{m})$-efficient implementation of the presented algorithm based on smart contract. Meanwhile, the features of the blockchain make sure all the results returned by smart contract must be correct, which means the invoking T makes have no need to be verified. Once U is dishonest, the fault will be detected by the blockchain network with a probability of 1. □

3.4 Efficiency Comparison

In this subsection, we compare efficiency and checkability of our algorithm with those of the previous algorithms based on single server, where efficiency is measured at computational cost for the outsourcer and the server while checkability is measured at the verification probability of the computing results from the server.

In Table 1, from the aspect of checkability and efficiency, we give the performance comparison for the outsourcer in the algorithms based on single server. In Table 2, also from the aspect of checkability and efficiency, we give the performance comparison for the single server in these algorithms. As shown in [15, 24] and Remark 1, to make the security of all algorithms being on the same level, we set $c=r=4$, $k=l=29$, $b'=33$, $b=53$, where χ, t_1, t_2 are all random number which is larger than 2^{64}.

According to Table 1, we can get the following conclusions. Firstly, compared with the algorithms in [15,16,19, 24], it is obvious that the presented algorithm is superior both in efficiency and checkability. Secondly, compared with the algorithms in [29], the outsourcer T in our algorithm has no need to execute the operation of modular inversions (MInvs) though T needs a little more modular multiplications (MMs). However, the outsourcer needs to compute MInvs for 9 times in the algorithm of [29]. It is well known that MInv has much higher computation complexity compared to MM, so the presented algorithm still has a higher efficiency than the algorithm in [29].

From Table 2, we can get the following conclusions. Firstly, compared with the algorithms in [16, 29], our algorithm needs less computing afford for the server and has higher checkability. Secondly, compared with the algorithms in [19, 24], there are more queries to the server in our algorithms. However, the checkability of the algorithms in [19, 24] are both smaller than 1. In another word, they are not sure for the outsourcing result and it is still possible for the server to cheat the outsourcer. Thirdly, compared with the algorithms in [15], there are still more queries to the server in our algorithms. Nevertheless, the queries to U in [15] is for exponential computation while the queries to U in our algorithm is for modular exponential computation. It is also known to us that exponential computation has much higher computational cost compared to that of modular exponentiation computation. That is to say, the computing efficiency for the server in our algorithm is still higher than that of [15] while keeping the checkability equivalence.

We will give the details of performance comparison in the experiments in 4.2. For the outsourcer, the computation efficiency in our algorithm is almost 2 times higher than that in [15]. And for the server, the computation efficiency in our algorithm is almost 5 times higher than that in [15]. Therefore, with the help of smart contract, there is no need for T to verify the results of the outsourcing computation, which greatly decreases the computation complexity of the outsourcing algorithms, so the presented algorithm based on smart contract has higher checkability and efficiency compared with the previous algorithms both for the outsourcer and the server.

4 Experimental Evaluation

In this section, we make a simulation experiment on Hyperledger Fabric for the outsourcing of modular exponentiation based on smart contract. Meanwhile, we give a comparison of computation performance between our outsourcing algorithm and direct computation for MExp, and the results show the proposed algorithm simultaneously improves the computation efficiency and the checkability. We also make the comparison of computation performance between our outsourcing algorithm and the previous algorithms based on single server from the aspect of the outsourcer and the server respectively, which shows that our algorithm has the best performance in these algorithms.

4.1 Realization

Hyperledger is an enterprise alliance blockchain platform, which was launched by the Linux foundation in 2015 to promote blockchain digital technology and transaction verification. It also makes blockchain technology can be widely used in other scenarios besides cryptocurrency. At the very beginning of the project, IBM, Jpmorgan Chase, Cisco, Intel and other technology and financial giants have joined it. As the representative of the third generation of blockchain, Hyperledger makes blockchain technology to be applied in business environment indeed.

Table 1: Efficiency comparison for the outsourcer among the algorithms with single server

	[24]	[27]	[15]	[5]	[21]	ours
MM	$12+1.5\log\chi$ ≥ 108	$1.5\log r+1.5\log t_1$ $+1.5\log t_2+15\geq 210$	$l+k+8\log c$ $+38=112$	22	$2b'+16=82$	$b+\lvert A\rvert$ $+5=59$
MInv	4	6	1	9	1	0
Rand	6	6	5	12	1	1
Checkability	0.5	0.991	≈ 0.917	≈ 1	1	1

Table 2: Efficiency comparison for the server among the algorithms with single server

	[24]	[27]	[15]	[5]	[21]	ours
Queries to U(for MExp)	4	4	$l+k+1=59$	170	0	b+2=55
Queries to U(for Exp)	0	0	0	0	b'+7=40	0
Checkability	0.5	0.991	≈ 0.917	≈ 1	1	1

Specially, as one of the most important sub-projects of Hyperledger, Fabric is widely used in blockchain application deployment. Same with the Ethereum, Hyperledger Fabric provide platform for developers to program smart contract according to their specially application requirements. And after deployment, the smart contract can be invoked to execute as the contents of it. Different from Ethereum, Hyperledger Fabric supports more conventional programming language, for example the Java, which is friendlier to developers [4].

In our experiment, smart contract is just the server for outsourcer to outsource the computation of modular exponentiation to. Specially, the computation of modular exponentiation will be the contents to be programmed to smart contract. After deployment, outsourcer can invoke the smart contract as the query to server to execute the computation of modular exponentiation.

We will mainly provide the following details to prove that the presented algorithm of outsourcing modular exponentiations truly has the higher efficiency than the previous algorithms proposed in [5, 15, 21, 24, 27] when they have the same level of security.

The parameters p and q are set similarly to [9], where p and q are all prime with the bit length of 512 and 160 respectively. In our experiment, there are specially set as:

$$p = fca682ce8e12caba26efccf7110e526db078b05edecbcd$$
$$1eb4a208f3ae1617ae01f35b91a47e6df63413c5e12ed0$$
$$899bcd132acd50d99151bdc43ee737592e17;$$
$$q = 962eddcc369cba8ebb260ee6b6a126d9346e38c5$$

The blockchain platform is Hyperledger Fabric and the programming language is Java. Therefore, the server is just the smart contract which is programmed and deployed in Hyperledger Fabric at version of 0.6 and simulated by the processors of Intel Core i5 at 2.3GHz with 8G RAM. The outsourcer is simulated by the processors of Intel Core i3 at 1.0GHz with 2G memory.

We simulate the outsourcing of computation of modular exponentiation when the base $u = 398812729285737041403682003227735175115965072869782930154223728815387998616677159083858143830985211011886947436393745905525757209133938731841748408187149 4$ and the exponent $a = 272122579386114794811291367966997255945069587330$.

The theoretical computation and outsourcing computation result equal as follows:
Outsourcing computation results:

$$127900628100902878828811051286114063670490517274354643311769029763171917797288457986733587209310575042484723473665941090868497579825258766061745497 94265465.$$

Theoretical Computation results:

$$127900628100902878828811051286114063670490517274354643311769029763171917797288457986733587209310575042484723473665941090868497579825258766061745497 94265465.$$

4.2 Efficiency Comparison with the Previous Algorithms

After the realization of our algorithm, we make the simulation experiment to show the computation performance of our algorithm. In the experiment, the smart contract is programmed and deployed in Hyperledger Fabric at version of 0.6 and simulated by the processors of Intel Core i5 at 2.3GHz with 8G RAM. The outsourcer and the server making direct computation are simulated by the processors of Intel Core i3 at 1.0GHz with 2G memory. The cloud server in the previous algorithms is simulated by the processors of Intel Core i7 at 3.4GHz with 4G RAM.

Firstly, we make the performance comparison between our algorithm and direct computation. Combining the outsourcer and the server in our algorithm with the server

making direct computation, we give the comparison of their computation efficiency, which is specifically reflected by computation time in Figure 3. It is clear that both the outsourcer and server in our algorithm have the lower computation consumption than the server which making direct computation for modular exponentiation. In another word, the outsourcing algorithm based on smart contract we presented greatly reduces the computing burden of the outsourcer and improves the efficiency of MExp computation.

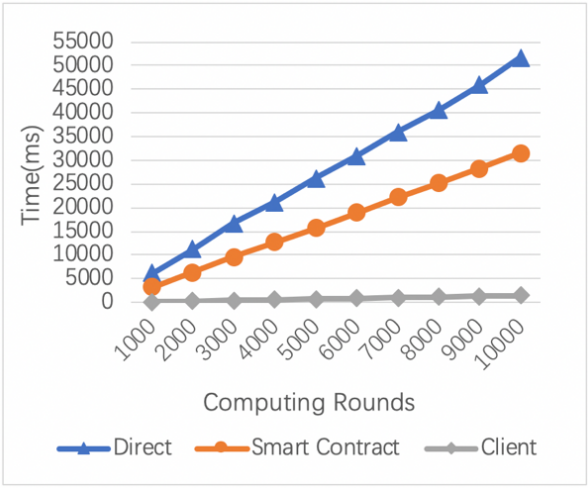


Figure 3: Simulation for the proposed algorithm with smart contract

Then, we make the performance comparison between our algorithm and the previous outsourcing algorithms based on single server. Specifically, in Figure 4 and Figure 5, we show the comparisons about the computation efficiency which is still reflected by computation time for the outsourcer and the server between our algorithm and the outsourcing algorithms proposed in [5, 15, 21, 24, 27]. As introduced in 3.4, to make security of these algorithms being at the same level, we set that $c = r = 4$, $k = l = 29$, $b' = 33$, $b = 53$, where χ, t_1, t_2 are all random numbers which are larger than 2^6.

As shown in Figure 4, it is obvious that computing time for the outsourcer in all of the algorithms grows linearly with the number of computing rounds approximately. And computing time for the outsourcer in our algorithm is always lower than those of the previous algorithms. Therefore, our algorithm has higher computing efficiency than all of the previous algorithms for the outsourcer. Specially, computing efficiency for the outsourcer in our algorithm is almost 2 times higher than that in [15].

From Figure 5, we can see that computing time for the server in all of the algorithms also grows linearly with the increase of computing rounds approximately. In detail, computing efficiency for the server in [24,27] is higher than that in our algorithm. However, their checkabilities are all smaller than 1 while the checkability in our algorithm keeps 1, which means security of the algorithms in [24, 27] is lower than ours. Compared to [5, 15], it is known

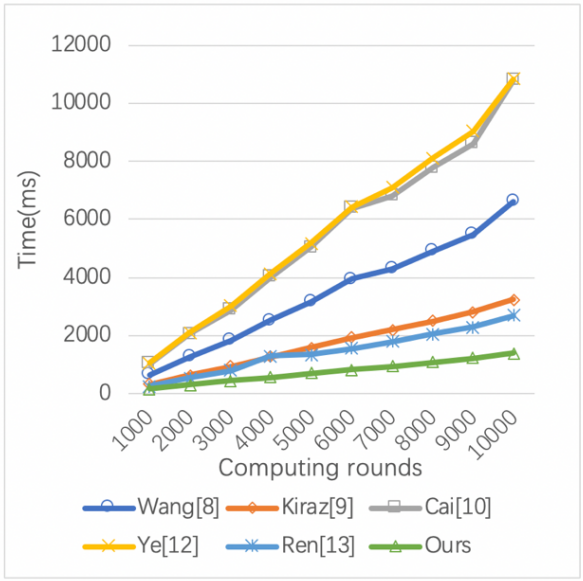


Figure 4: Time comparison of the outsourcer among the outsourcing algorithms with single server

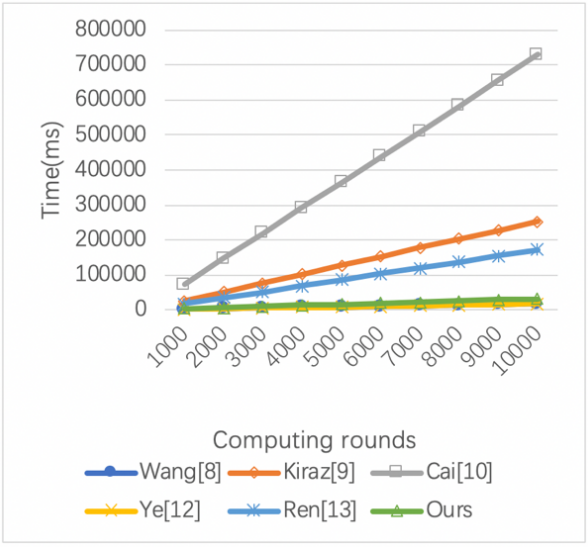


Figure 5: Time comparison of the server among the outsourcing algorithms with single server

to us that no matter the efficiency or the checkability, the presented algorithm is all higher than theirs. Then compared to [21], our algorithm has higher efficiency for the server than that in [21] while two algorithms have same checkability of 1. Specially, computing efficiency for the server in our algorithm is almost 5 times higher than that in [15].

In conclusion, combining checkability with computing efficiency for the outsourcer and the server, our algorithm has higher efficiency while keeps the outsourcing process secure. Therefore, our algorithm has better performance including security and efficiency compared with the outsourcing algorithm in [5, 15, 21, 24, 27].

5 Conclusions

We present an algorithm of outsourcing modular exponentiation based on smart contract with a checkability of 1 in this paper. We firstly give a security model in outsourcing computation based on smart contract. Then we propose a specific outsourcing algorithm and give its secure analysis in the security model. Finally, we make a simulation experiment to prove the correctness of the algorithm and show it truly decreases the computation cost for the direct computation of MExp and has higher efficiency than those of the previous algorithms based on single server at the same security level.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China (U1736120, 61572309, U1536109), and Natural Science Foundation of Shanghai (19ZR1419000).

References

[1] M. Atallah, K. Frikken, "Securely outsourcing linear algebra computations," in *Proceedings of the 5th ACM Symposium on Information*, pp. 48-59, 2010.

[2] Y. N. Aung, T Tantidham, "Review of Ethereum: smart home case study," in *Proceedings of 2nd International Conference on Information Technology*, 2017. (`https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8257877`)

[3] A. Baliga, N. Solanki, S. Verekar, *et al.*, "Performance characterization of hyperledger fabric," in *Proceedings of Crypto Valley Conference on Blockchain Technology*, pp. 65-74, 2018.

[4] L. Cai. X. Liang, Z. Yi, "The source code analysis of Hyperledger Fabric," *China Machine Press*, pp. 18-34, 2018.

[5] J. Cai, Y. Ren, C. Huang, "Verifiable outsourcing computation of modular exponentiations with single server," *International Journal of Network Security*, vol. 19, no. 3, pp. 449-457, 2017.

[6] X. Cai, Y. Ren, X. Zhang, "Privacy-protected deletable blockchain," *IEEE Access*, 2019. DOI 10.1109/ACCESS.2019.2962816.

[7] A. Chattopadhyay, A. Nag, K. Majumder, "Secure data outsourcing on cloud using secret sharing scheme," *International Journal of Network Security*, vol. 19, no. 6, pp. 912-921, 2017.

[8] D. Chaum, T. Pedersen, "Wallet databases with observers," in *Proceegings of 12th Annual International Cryptology Conference*, pp. 89-105, 1992.

[9] X. Chen, J. Li, and J. Ma, "New algorithms for secure outsourcing of modular exponentiations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2386-2396, 2014.

[10] M. Dijk, D. Clarke, B. Gassend, G. Suh, and S. Devadas, "Speeding up exponentiation using an untrusted computational resource," *Designs, Codes and Cryptography*, vol. 39, no. 2, pp. 253-273, 2006.

[11] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proceedings of 30th Annual Cryptology Conference*, pp. 465-482, 2010.

[12] P. Hegedus, "Towards analyzing the complexity landscape of solidity based ethereum smart contracts," *Technologies*, vol. 7, no. 1, pp. 1-16, 2018.

[13] S. Hohenberger, A. Lysyanskaya, "How to securely outsource cryptographic computations," in *Proceedings of 2nd Theory of Cryptography Conference*, pp. 264-282, 2005.

[14] W. Hsien, C. Yang, M. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133-142, 2016.

[15] M. Kiraz, O. Uzunkol, "Efficient and verifiable algorithms for secure outsourcing of cryptographic computations," *International Journal of Information Security*, vol. 15, no. 5, pp. 519-537, 2016.

[16] T. Matsumoto, K. Kato and H. Imai, "Speeding up secret computations with insecure auxiliary devices," in *Proceedinga of Conference on the Theory and Application of Cryptography*, pp. 497-506, 1988.

[17] P. Nguyen, I. Shparlinski, and J. Stern, "Distribution of modular sums and the security of server aided exponentiation," in *Proceedings of International Conference on Computer Science and Network Technology*, pp. 1-16, 1999.

[18] S. Nick, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, pp. 1-2, 1997.

[19] B. Parno, M. Raykova and V. Vaikuntanathan, "How to delegate and verify in public: Verifiable computation from attribute based encryption," in *Proceedings of 9th Theory of Cryptography Conference*, pp. 422-439, 2012.

[20] Y. Ren, M. Dong, G. Feng, "Fully verifiable algorithm for outsourcing multiple modular exponentiations with single cloud server," *IEICE Transac-*

tions on Fundamentals of Electronics Communications and Computer Sciences, vol. 101, no. 3, pp. 608-611, 2018.

[21] Y. Ren, M. Dong, Z. Qian, *et al.*, "Efficient algorithm for secure outsourcing of modular exponentiation with single server," *IEEE Transactions on Cloud Computing*, 2018. DOI: 10.1109/TCC.2018.2851245.

[22] D. R. Stinson, *Cryptography Theory and Practice*, 2016. ISBN 13: 978-1584885085.

[23] S. Thokchom, D. Saikia, "Privacy preserving and public auditable integrity checking on dynamic cloud data," *International Journal of Network Security*, vol. 21, no. 2, pp. 221-229, 2019.

[24] Y. Wang, Q. Wu, D. S. Wong, "Securely outsourcing exponentiations with single untrusted program for cloud storage," in *Proceedings of 19th European Symposium on Research in Computer Security*, pp. 326-343, 2014.

[25] Y. Wu, S. Tang, B. Zhao. *et al.*, "BPTM: Blockchain-based privacy-preserving task matching in crowdsourcing," *IEEE Access*, vol. 7, pp. 45605-45617, 2019.

[26] J. Ye, X. Chen, and J. Ma, "An improved algorithm for secure outsourcing of modular exponentiations," in *Proceedings of 29th International Conference on Advanced Information Networking and Applications*, pp. 73-76, 2015.

[27] J. Ye, J. Wang, "Secure outsourcing of modular exponentiation with single untrusted server," in *Proceedings of 18th International Conference on Network-Based Information Systems*, pp. 643-645, 2015.

[28] W. Zhao, "An initial review of cloud computing services research development," in *Proceegings of International Conference on Multimedia Information Networking and Security*, pp. 324-328, 2010.

[29] H. Zhu, Y. Zhang, Z. Li, "A novel and provable authenticated key agreement protocol with privacy protection based on chaotic maps towards mobile network," *International Journal of Network Security*, vol. 18, no. 1, pp. 116-123, 2016.

Danting Xu is a master degree candidate in School of Communication and Information Engineering at Shanghai University, China. She was awarded a B.S. degree in communication engineering in 2017 from Shanghai University, China. Her research interests include secure outsourcing computing, and network security.

Yanli Ren is a professor in School of Communication and Information Engineering at Shanghai University, China. She was awarded a M.S. degree in applied mathematics in 2005 from Shanxi Normal University, China, and a PhD degree in computer science and technology in 2009 from Shanghai Jiao Tong University, China. She has published more than 70 papers on international journals and conferences. Her research interests include applied cryptography, secure outsourcing computing, network security and blockchain security.

Xiangyu Li is a master degree candidate in School of Communication and Information Engineering at Shanghai University, China. He was awarded a B.S. degree in communication engineering in 2018 from Shanghai University, China. His research interests include secure outsourcing computing, and network security.

Guorui Feng received the B.S. and M.S. degree in computational mathematic from Jilin University, China, in 1998 and 2001 respectively. He received Ph.D. degree in electronic engineering from Shanghai Jiao-tong University, China, 2005. From January 2006 to December 2006, he was an assistant professor in East China Normal University, China. During 2007, he was a research fellow in Nanyang Technological University, Singapore. Now he is with the school of communication and information engineering, Shanghai University, China. His current research interests include image processing, image analysis and computational intelligence.

A Lightweight Anonymous Mobile Payment Scheme for Digital Commodity in Cloud Computing Service

Baoyuan Kang, Jianqi Du, Yanbao Han, and Kun Qian
(Corresponding author: Baoyuan Kang)
School of Computer Science and Technology, Tianjin Polytechnic University
Tianjin, 300387, China
(Email: baoyuankang@aliyun.com; 2045409630@qq.com; 516283720@qq.com; 923535956@qq.com)
(Received June 11, 2019; Revised and Accepted Dec. 22, 2019; First Online Feb. 1, 2020)

Abstract

With the development of internet technology, mobile payment has entered our life and brought great convenience to daily buying. But, prevailing payment systems do not provide privacy-protection for customers. Some existing payment schemes have high computation cost. In this paper, combining efficient cloud computing service and e-cash schemes providing privacy protection, we propose a lightweight anonymous mobile payment scheme for digital commodity. Compared with some existing payment schemes, the proposed scheme not only satisfies fairness and common safety requirements, but also efficiently reduces the computational burden of merchants with a lot of businesses.

Keywords: Anonymity; Cloud Computing; Digital Commodity; Fairness; Mobile Payment

1 Introduction

With the development of internet technology, mobile payment has entered our life and brought great convenience to daily buying [10, 21, 22]. But, prevailing payment systems do not provide privacy-protection for customers. Such systems can reveal not only purchases information of customers, but also customer's details including their names, emails, postal addresses [16].

To address the aforementioned issues, only anonymous payment schemes that protect personal information can be used. Relevant research works [5, 18, 20, 24] have been conducted in recent years, and their concerned issues focus on security, privacy in mobile payment. But high computation cost makes these systems unsuitable for resource-constrained mobile devices. Recently, based on signcryption algorithms, Cao *et al.* [2] proposed a strong anonymous mobile payment scheme. Unfortunately, in their scheme merchants have advantages over customers, since customers should give money at once to merchants. Even though Cao *et al.*'s scheme has revocation function for settling disputes, consumers' identity information will be exposed in revocation phase. This contradicts the requirement of anonymity. Also Cao *et al.*'s scheme [18] cannot efficiently resolve disputes and include many time-consuming bilinear pairing computations.

In fact, e-cash [7, 9, 11, 12, 14, 15, 23] schemes are ideal approach to achieve anonymous payment. Kang et al [7] proposed an e-cash anonymous payment scheme with low computation cost compared with many e-cash payment schemes. But, their scheme is not suitable to mobile payment since there are also time-consuming bilinear pairing computations.

In recent years, with the development of network technology, digital economy and cloud services are becoming more and more popular [8, 13, 17]. Many people buy digital goods and digital services through the Internet. Cloud services have accelerated such business growth with their powerful computing and processing capabilities. In this paper, combining efficient cloud service and e-cash providing privacy protection, we propose a lightweight anonymous mobile payment scheme for digital commodity. To merchants with a lot of businesses, the proposed scheme can efficiently reduce their computational burden.

The rest of the paper is organized as follows. In Section 2, we give the system and security model. In Section 3, we give some parameters and briefly review Ting *et al.*'s signcryption scheme used in the proposed scheme. Our scheme is proposed in Section 4. Security cryptanalysis and comparisons are given in Section 5. Finally, the article is concluded in Section 6.

2 The System and Security Models

This section introduces the system and security models of our proposed mobile payment scheme for digital commod-

ity in cloud service environment. It will describe the role of participants in the proposed scheme and put forward security objectives.

2.1 The System Model

The system model of our proposed mobile payment scheme is shown in Figure 1. It is consists of four entities:

1) A customer who with a smart phone can access the Internet to buy digital commodities or digital services using e-cash from a merchant.

2) A merchant who sells various digital commodities or digital services.

3) A bank that generates valid e-cash for the customer and accepts e-cash deposit from the merchant.

4) A cloud service platform (CSP) which provides payment service and solves disputes between the customer and the merchant.

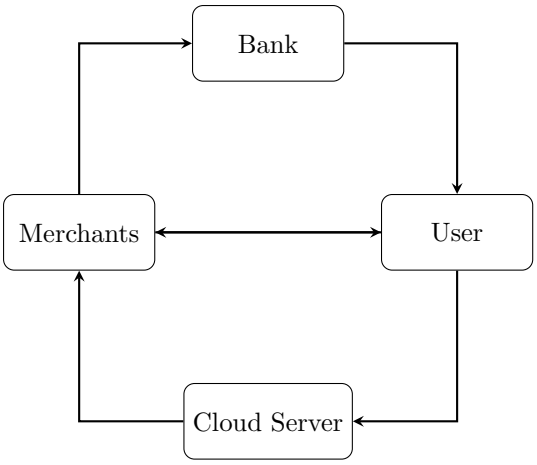


Figure 1: The system model

2.2 The Security Model

In a mobile payment scheme for digital commodity in cloud service environment, the bank and the cloud service platform CSP are honest but curious. They generally strictly execute the terms of the scheme. But, they are also interested in the customer's identity information. The customer and the merchant may deceive the bank and CSP for their benefit. The customer may forge an e-cash to cheat the merchant and the bank. The merchant may cheat the customer in digital commodities or digital services.

In view of the above-mentioned models, our security objectives are threefold:

1) Anonymity: The proposed scheme will protect the customer's identity information. The bank cannot know who ever withdrawn the e-cash deposited by a merchant. CSP and the merchant cannot know the real identity of the customer who conducts a transaction with the merchant through CSP.

2) Unforgeability: No one but the bank can generate valid e-cash. One who personates others or forges messages cannot pass validity verifications.

3) Fairness: Neither the customer nor the merchant has advantages over the other in dealing. Even a dispute arises, it can be dealt with fairly through CSP.

3 Preliminary

In this section we give some parameters and notations used in our scheme. We also introduce Ting *et al.*'s signcryption scheme [19] used in our scheme.

3.1 The Notations

The notations used in this article are shown in Table 1.

Table 1: The notations

Symbol	Description
G_1	An additive group of prime order q
P	A generator of G_1
PKG	The Private Key Generator
w	PKG's private key
P_{pub}	PKG's public key
H_1, H_2, H_3	Three hash functions
U	A customer
ID_U	Identity of the customer U
M	A merchant
ID_M	Identity of the merchant M
x_B, PK_B	Private key and public key of the bank
x_{CSP}, PK_{CSP}	Private key and public key of the cloud service platform (CSP)
N	Denomination of e-cash
DC	A digital commodity's name or content
$Cert_{DC}$	The certificate of the digital commodity DC

3.2 Ting *et al.*'s Signcryption Scheme

In a signcryption scheme there are a sender and designated receiver. The sender can achieve encryption and signature in a logical single step. So, a signcryption scheme can efficiently provide confidentiality and authentication. Ting *et al.*'s signcryption scheme is indistinguishable against adaptive chosen-ciphertext attacks under the computational Diffie'Hellman assumption, and is unforgeable against adaptive chosen-message attacks under the elliptic curve discrete logarithm assumption. Following is the description of Ting *et al.*'s signcryption scheme [19].

Let G_1 be an additive group of prime order q and P be a generator of G_1. The Private Key Generator (PKG) randomly chooses a number $w \in Z_q$ as its master private key and computes the corresponding public key $P_{pub} = wP$. PKG also chooses three one-way hash functions:

$H_1 : G_1 \times Z_q \to Z_q$,

$H_2 : G_1 \times G_1 \times G_1 \to Z_q$,

$H_3 : Z_q \times G_1 \times G_1 \times G_1 \times Z_q \times Z_q \to Z_q$,

then publishes system parameters$\{P, P_{pub}, q, H_1, H_2, H_3\}$.

For a user U with his identity ID_U, PKG randomly chooses $r_U \in Z_q$, computes

$R_U = r_U P$, $D_U = r_U + wH_1(R_U, ID_U) \mathrm{mod} q$,

and securely returns $sk_U = (R_U, D_U)$ to U as U 's private key.

Signcrypt: Given a sender's private key sk_U and a receiver's public key PK_T (correspondingly private key is x_T and $PK_T = x_T P$), the algorithm firstly chooses randomly two numbers $b_1, b_2 \in Z_q$, and computes

$$B_1 = b_1 P, B_2 = b_1(PK_T), d = b_2 D_U \bmod q.$$

Then the algorithm computes

$$\begin{aligned} c &= m \oplus H_2(R_U, B_1, B_2), \\ h &= H_3(m, R_U, B_1, B_2, d, c), \\ v &= (h + b_1)^{-1} {b_2}^{-1} \bmod q. \end{aligned}$$

The signcrypted ciphertext for the message m is $\sigma = (c, R_U, B_1, d, v)$.

Unsigncrypt: Given a ciphertext $\sigma = (c, R_U, B_1, d, v)$, the sender's identity ID_U and the receiver's private key x_T, this algorithm computes

$$\begin{aligned} B_2 &= x_T B_1, \\ m &= c \oplus H_2(R_U, B_1, B_2), \\ h &= H_3(m, R_U, B_1, B_2, d, c), \end{aligned}$$

and checks the validity of message by the following equation:

$$vd(B_1 + hP) = R_U + H_1(R_U, ID_U) P_{pub}.$$

4 The Proposed Scheme

Suppose that a customer U and a merchant M have their account in advance at a bank. When U wants to buy a digital commodity from M, U firstly browses M's website, and gets the price N of his wanted digital commodity. Then, U withdraws a N denomination e-cash at the bank and pays the e-cash to the merchant M through the cloud service platform CSP. Finally, U obtains his wanted digital commodity.

The proposed scheme consists of four phases: the withdrawal phase, the payment phase, the deposit phase and the adjudication phase. In the proposed scheme, the customer U and the merchant M have identity ID_U and ID_M, respectively. The customer U has his id-based private key (R_U, D_U) generated by PKG. Here $D_U = r_U + wH_1(R_U, ID_U) \mathrm{mod} q$, $R_U = r_U P$, w is PKG's private key. The bank and CSP have their private/public key pairs $x_B / PK_B = x_B P$, $x_{CSP} / PK_{CSP} = x_{CSP} P$, respectively.

4.1 The Withdrawal Phase

In the withdrawal phase, we use the idea of Ting *et al.* signcryption scheme [19] and blind signatures [9] to make the customer U to securely get an e-cash from the bank.

Firstly, U sends a request information (including U's identity ID_U and account Information AI_U) for an e-cash to the bank. When the request information passes the verifications, the bank sends an e-cash to U by the following steps. We also depict the withdrawal phase in Figure 2.

1) The bank chooses element $a \in Z_q^*$ computes $A = aP$ and sends A to U.

2) U receives A and generates his pseudonym $PID_U = H_1(ID_U, z)$, where z is a random number in Z_q^*. Then, U randomly chooses $b_1, b_2 \in Z_q$, and computes

$$\begin{aligned} B_1 &= b_1 A, B_2 = b_1(PK_B), d = b_2 D_U \bmod q. \\ m &= ID_U || N || b_1^{-1} b_2 H_1(PID_U, ID_M, B_1), \\ c &= m \oplus H_2(R_U, B_1, B_2), \\ h &= H_3(m, R_U, B_1, B_2, d, c), \\ v &= (h + b_1)^{-1} b_2^{-1} \bmod q, \end{aligned}$$

and sends $\sigma = (c, R_U, B_1, d, v)$ to the bank.

3) On receiving $\sigma = (c, R_U, B_1, d, v)$, the bank computes

$$\begin{aligned} B_2 &= a^{-1} x_B B_1, \\ m &= c \oplus H_2(R_U, B_1, B_2), \\ h &= H_3(m, R_U, B_1, B_2, d, c), \end{aligned}$$

and extracts the customer's identity information from the message m. Then the bank checks the following equation:

$$vd(a^{-1} B_1 + hP) = R_U + H_1(R_U, ID_U) P_{pub}.$$

4) When the equation holds, the bank extracts N and $b_1^{-1} b_2 H_1(PID_U, ID_M, B_1)$ from the message m and computes

$$S' = a^{-1} b_1^{-1} b_2 H_1(PID_U, ID_M, B_1) H_2(N, t) x_B \bmod q.$$

Where t is the deadline for the e-cash. Then the bank sends (N, t, S') to U.

5) On receiving (N, t, S'), U computes $S = b_2^{-1} S'$, and checks the following equation

$$SB_1 = H_1(PID_U, ID_M, B_1) H_2(N, t)(PK_B).$$

If the equation holds, U obtains a valid e-cash $(PID_U, ID_M, B_1, S, N, t)$.

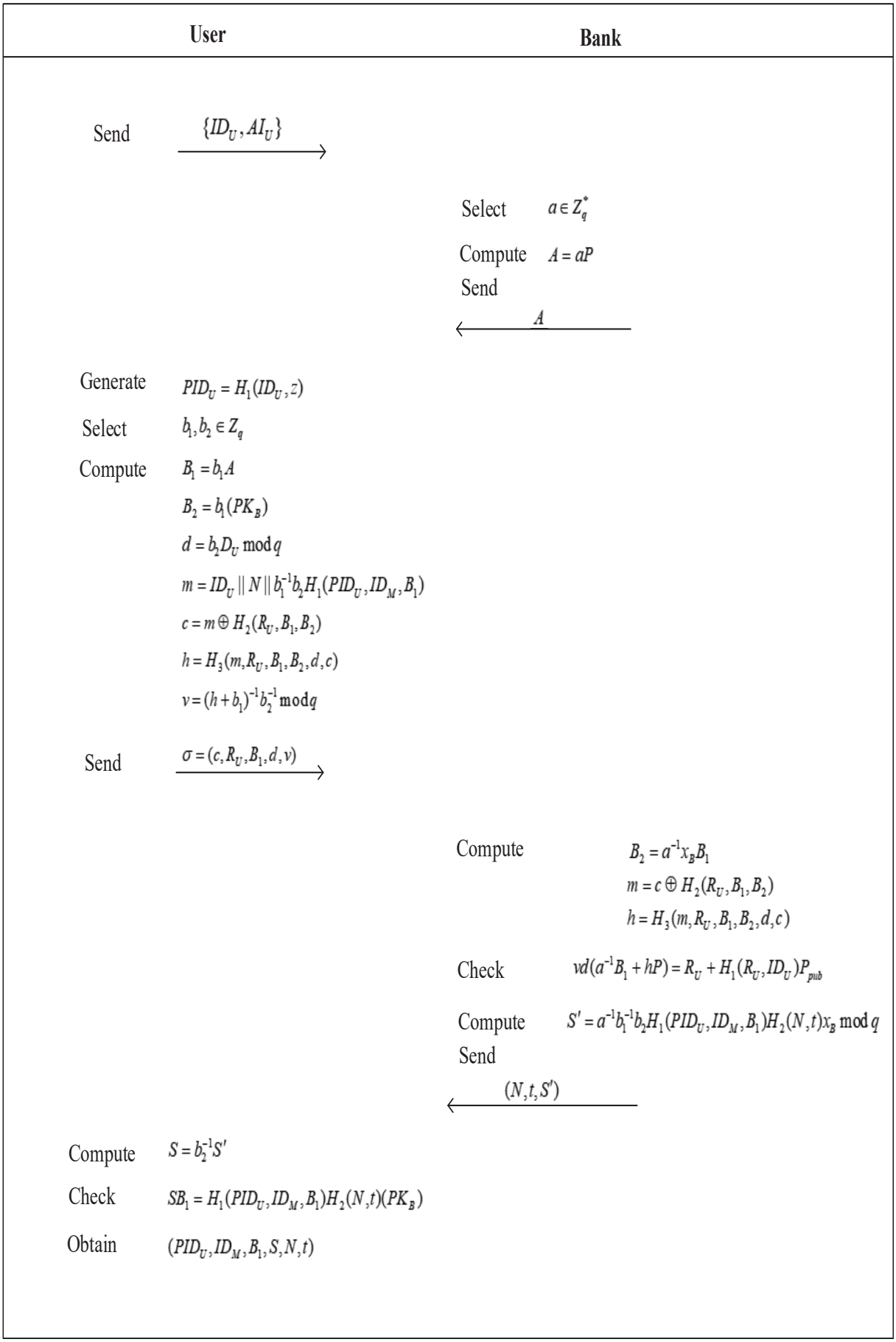


Figure 2: The withdrawal phase

4.2 The Payment Phase

During the payment phase, the customer U pays the e-cash to the merchant M though the cloud service platform CSP. We depict this phase in Figure 3. Following is the detailed steps.

1) U chooses a random number $y \in Z_q^*$ and computes

$$Y = yP,\ F = (S + H_1(y(PK_{CSP}), DC, Y)) \bmod q,$$

and send $(PID_U, ID_M, B_1, F, N, t, Y, DC)$ to CSP. Here DC only denotes the wanted digital commodity's name.

2) On receiving $(PID_U, ID_M, B_1, F, N, t, Y, DC)$, CSP obtains by the following computation

$$S = (F - H_1(x_{CSP}Y, DC, Y)) \bmod q,$$

and checks whether the equation

$$SB_1 = H_1(PID_U, ID_M, B_1)H_2(N, t)(PK_B),$$

holds or not. If it holds, and there is not $(PID_U, ID_M, B_1, F, N, t, Y, DC)$ record in CSP-list, CSP chooses a random number $j \in Z_q^*$, and computes

$$\begin{aligned} J &= jP, \\ L_1 &= (j + H_2(PID_U, ID_M, S, N, t)x_{CSP}) \bmod q, \\ L_2 &= (j + H_2(PID_U, ID_M, DC, N, Y, t)x_{CSP}) \bmod q. \end{aligned}$$

Then, CSP sends (J, L_1) to U, sends $(PID_U, ID_M, DC, N, Y, t, J, L_2)$ to M and records $(PID_U, ID_M, B_1, F, N, t, Y, DC)$ in the CSP-list.

3) Upon receiving $(PID_U, ID_M, DC, Y, t, J, L_2)$, the merchant M firstly checks the equation

$$L_2P = J + H_2(PID_U, ID_M, DC, N, Y, t)Y_{CSP}.$$

If the equation holds, M encrypts the digital commodity DC with Y, and sends $E_Y(DC)$ to U. Here E is a public encryption algorithm.

4) Upon receiving $E_Y(DC)$, U decrypts it with the corresponding secret value y, gets DC and checks it. If DC is valid, U chooses a random number $i \in Z_q^*$, computes

$$I = iP,\ Q = (y + iH_2(PID_U||DC)) \bmod q,$$

and sends (I, Q) to M.

5) On receiveing (I, Q), M checks the following equation

$$QP = Y + H_2(PID_U||DC)I.$$

If the equation holds, M sends $E_Y(Cert_{DC})$ to U. Meanwhile, M sends $(PID_U, ID_M, N, Y, I, Q, J, L_2)$ to CSP.

6) On receiveing $E_Y(Cert_{DC})$, U decryptes it, gets the certificate $Cert_{DC}$ of digital commodity DC and check the certificate.

7) On receiveing $(PID_U, ID_M, N, Y, I, Q, J, L_2)$, CSP checks the following equations

$$\begin{aligned} L_2P &= J + H_2(PID_U, ID_M, DC, N, Y)Y_{CSP}, \\ QP &= Y + H_2(PID_U||DC)I. \end{aligned}$$

If the equations holds, CSP sends the e-cash $(PID_U, ID_M, B_1, S, N, t)$ to M. After the expiry date, CSP delete $(PID_U, ID_M, B_1, S, N, t)$ from the CSP-list.

4.3 The Deposit Phase

In this phase, when the bank obtains an e-cash from the merchant, the bank deposits money into the merchat's account.

1) Before the deadline t of the e-cash $(PID_U, ID_M, B_1, S, N, t)$, the merchant M chooses a random number $k_1 \in Z_q^*$, computes

$$K_1 = k_1P,\ O_1 = (S + H_1(k_1(PK_B), K_1)) \bmod q,$$

and sends $(PID_U, ID_M, B_1, O_1, N, t, K_1)$ to the bank.

2) On receiving $(PID_U, ID_M, B_1, O_1, N, t, K_1)$, the bank computes

$$S = (O_1 - H_1(x_BK_1, K_1)) \bmod q,$$

and checks whether the equation

$$SB_1 = H_1(PID_U, ID_M, B_1)H_2(N, t)(PK_B),$$

holds or not. If the equation holds, the bank deposits N amount money into the merchant M's account.

4.4 The Adjudication Phase

In the Step 4 of the payment phase, if the customer U finds the commodity DC is not valid, or DC is valid, but U does not go on the terms of the scheme, then the merchant cannot receive (I, Q) from U in time. Whatever the case may be, in the near deadline t, the merchant M chooses a random number $k_2 \in Z_q^*$, computes

$$\begin{aligned} V &= DC||Cert_{DC}, \\ K_2 &= k_2P, \\ O_2 &= (V + H_1(k_2(PK_{CSP}), K_2)) \bmod q, \end{aligned}$$

and sends $(PID_U, ID_M, B_1, O_2, N, t, K_2)$ to CSP.

On receiving $(PID_U, ID_M, B_1, O_2, N, t, K_2)$, CSP computes

$$V = (O_2 - H_1(x_{CSP}K_2, K_2)) \bmod q,$$

obtains DC, $Cert_{DC}$ and checks them. When DC and $Cert_{DC}$ are valid, CSP sends the e-cash to the merchant M, and sends $E_Y(DC||Cert_{DC})$ to the customer U.

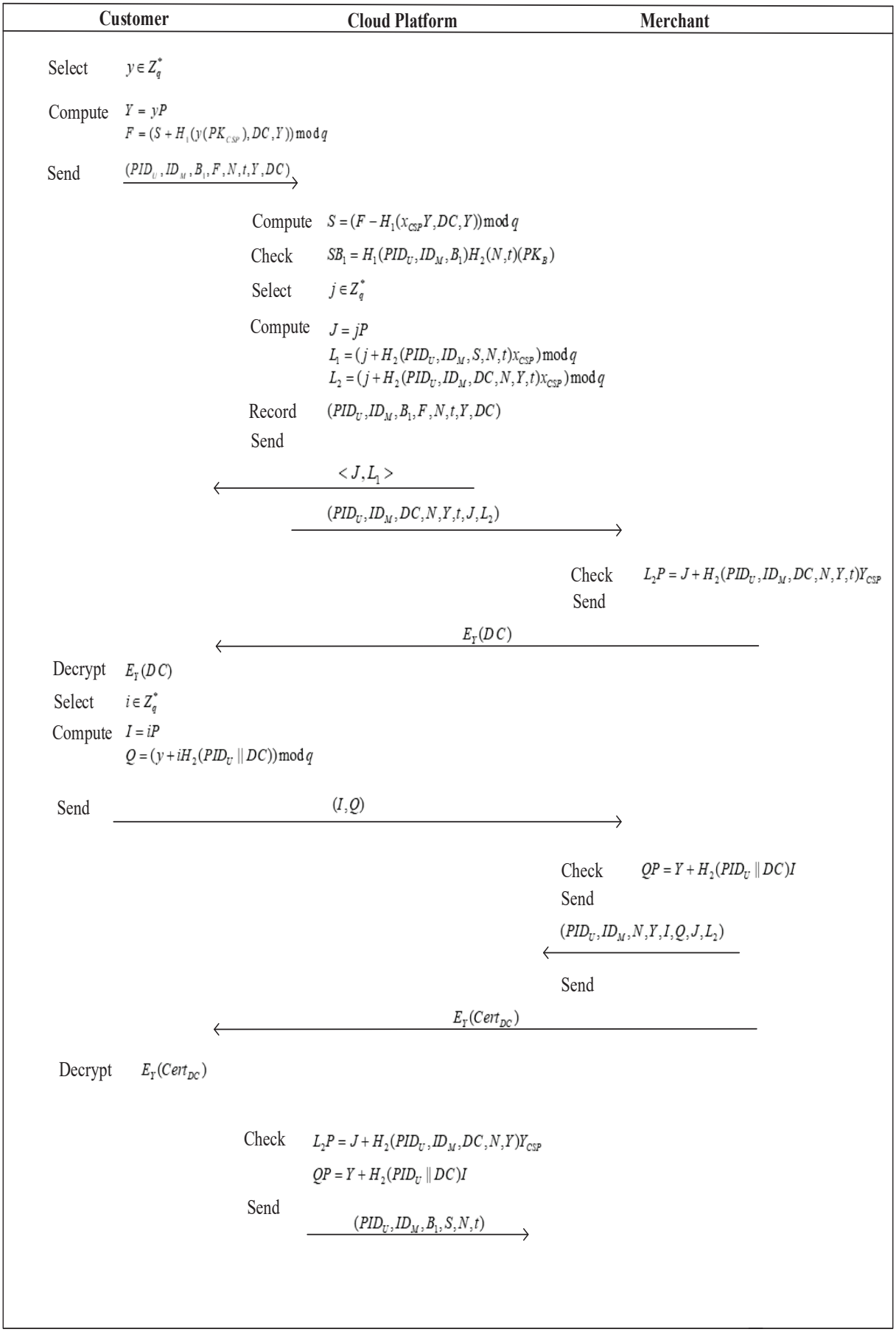


Figure 3: The payment phase

5 Security Analysis and Comparisons

In this section, we evaluate the security of the proposed scheme, including anonymity, unforgeability and fairness analysis. We also show the comparison results of the proposed scheme with some existing schemes in security and computational efficiency.

5.1 Security Analysis

We analyze the security of the proposed scheme from the following three aspects.

1) Anonymity. In the whole payment phase, only the customer U's pseudonym $PID_U = H_1(ID_U, z)$ directly related to the real identity is used. But, the real identity ID_U of the customer U is protected by hash function, any one cannot obtain ID_U from the pseudonym PID_U. So, in the whole payment phase neither CSP nor the merchant M knows the real identity of the customer U.

   Moreover, since we use signcryption and blind signature technology in the withdrawal phase and only pseudonym PID_U appears in the e-cash, when the merchant deposits the e-cash to the bank, the bank cannot track the real identity of the customer who withdraw the e-cash from the bank.

   In summary, our proposed scheme satisfies the anonymity secure property.

2) Unforgeability. In the payment phase and deposit phase, since all transmitted vital messages, such as

$$\begin{aligned} F &= (S + H_1(y(PK_{CSP}), DC, Y)) \bmod q, \\ L_1 &= (j + H_2(PID_U, ID_M, S, N, t)x_{CSP}) \bmod q, \\ Q &= (y + iH_2(PID_U||DC)) \bmod q, \\ O_1 &= (S + H_1(k_1(PK_B), K_1)) \bmod q, \end{aligned}$$

   are generated by secure ElGamal-Type encryption and signature technology [1] using corresponding public key in encryption and private key or secret value of the senders in signatures, and all transmitted messages must pass the verifications, anyone cannot personate others to forge valid messages.

   In the withdrawal phase, the bank generates an e-cash by using its private key. So, no one can produce valid e-cash except the bank. Moreover, on one can personate the customer U to withdraw an e-cash, because the secure Ting *et al.* signcryption scheme [19] is used in the withdrawal phase.

   Therefore, our proposed scheme satisfies the unforgeability secure property.

3) Fairness. In the proposed scheme when a customer wants to buy digital commodity from a merchant, the customer sends his e-cash to the CSP instead of giving the e-cash to the merchant directly. Only after the merchant sends the valid wanted digital commodity to the customer and shows the receipt information (I, Q) from the customer to CSP, CSP sends the e-cash to the merchant. So, in the proposed scheme the merchant has not advantages over the customer.

On the other hand, when the merchant receives the valid purchase information $(PID_U, ID_M, DC, Y, t, J, L_2)$ from CSP, the merchant encrypts the digital commodity and sends to the customer. Here, we note that the merchant does not send the certificate $Cert_{DC}$ with the commodity DC. Only after the merchant receives the receipt information (I, Q) from the customer, the merchant sends the certificate $Cert_{DC}$ to the customer. Even if a unkind customer does not send the receipt information to the merchant, the merchant also can obtain the e-cash in the adjudication phase. So, in the proposed scheme the customer has not virtual advantages over the merchant.

To sum up above mentioned two aspects, neither a customer nor a merchant has advantages over each other in dealing. Even a dispute arises, it can be dealt with fairly through CSP. The proposed scheme satisfies the fairness property.

5.2 Comparisons

In this section, the comparisons of the proposed scheme with schemes [2, 3, 5, 6] are shown. The comparison results of the security features and computation costs are shown in Table 2, Table 3, respectively. From Table 2, our proposed scheme satisfies anonymity, unforgeability and fairness, it is superior to the schemes [2, 3, 5] in security properties. Since there are on concrete signature/verification algorithm in the schemes [3, 6], we see once signature/verification in the two schemes as once encryption/decryption operation for the comparison of computation cost. We also omit the statistics of multiplication operation and inverse operation in the two schemes. From Table 3, the total computation cost of Cao *et al.* scheme [2], Chaudhry *et al.* scheme [3] and Kang *et al.* scheme [6] are $T_1 = 8X+14H+6P+6(E/D)$, $T_2 = 6X+2H+11(E/D)$ and $T_3 = 6X+3H+14(E/D)$, respectively. But, the total computation cost of our proposed scheme is $T_4 = 30X + 23H + 2(E/D)$. According to [4], $T_1 \approx 8662X + 6(E/D)$, $T_2 \approx 52X + 11(E/D)$, $T_3 \approx 75X + 14(E/D)$, $T_4 \approx 53X + 2(E/D)$. So, our proposed scheme is also superior to the schemes [2, 3, 6] in computation cost.

Table 2: Comparison of security (F1: Anonymity, F2: Unforgeability, F3: Fairness)

	F1	F2	F3
Isaac *et al.* scheme [5]	No	Yes	No
Cao *et al.* scheme [2]	Yes	Yes	No
Chaudhry *et al.* scheme [3]	No	Yes	No
Our scheme	Yes	Yes	Yes

Table 3: Comparison of computation costs (X: Scalar multiplication in Groups, H: Hash Calculation, P: Bilinear pairing operation, (E/D):Encryption/decryption operation)

Schemes	Ticket(e-cash)issuance	Payment phase	Deposit phase
Cao *et al.* scheme [2]	$8X + 10H + 2P$	$2H + 2P + 4(E/D)$	$2H + 2P + 2(E/D)$
Chaudhry *et al.* scheme [3]	$2X + H + 4(E/D)$	$2X + H + 6(E/D)$	$2X + (E/D)$
Kang *et al.* scheme [6]	$2X + H + 3(E/D)$	$4X + H + 9(E/D)$	$H + 2(E/D)$
Our scheme	$10X + 8H$	$15X + 11H + 2(E/D)$	$5X + 4H$

6 Conclusions

In this paper, we propose a lightweight anonymous mobile payment scheme for digital commodity. The scheme is very suitable to merchants having a lot of businesses, since the proposed scheme can efficiently reduce the computational burden of such merchants by cloud services. We discuss the security of the proposed scheme in anonymity, unforgeability and fairness. Furthermore, we compare our scheme with some existing mobile payment schemes. The proposed scheme is superior to some existing schemes in security and computational efficiency and suitable for resource-constrained mobile payment devices. Next, we should try our best to reduce the calculation cost significantly.

Acknowledgments

We would like to thank the reviewers for their helpful comments. This work is supported by the Applied Basic and Advanced Technology Research Programs of Tianjin (No. 15JCYBJC15900) and the National Natural Science Foundation of China (No. 61972456).

References

[1] M. Burmester, Y. Desmedt, H. Doi, M. Mambo, E. Okamoto, M. Tada, and Y. Yoshifuji, "A structured elgamal-type multisignature scheme," in *International Workshop on Public Key Cryptography*, pp. 466–483, 2000.

[2] C. Cao and X. Zhu, "Strong anonymous mobile payment against curious third-party provider," *Electronic Commerce Research*, vol. 19, no. 3, pp. 501–520, 2019.

[3] S. A. Chaudhry, M. S. Farash, H. Naqvi, and M. Sher, "A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography," *Electronic Commerce Research*, vol. 16, no. 1, pp. 113–139, 2016.

[4] C. I. Fan, W. Z. Sun, and V. S. M. Huang, "Provably secure randomized blind signature scheme based on bilinear pairing," *Computers & Mathematics with Applications*, vol. 60, no. 2, pp. 285–293, 2010.

[5] J. T. Isaac, S. Zeadally, and J. S. Cámara, "A lightweight secure mobile payment protocol for vehicular ad-hoc networks (VANETs)," *Electronic Commerce Research*, vol. 12, no. 1, pp. 97–123, 2012.

[6] B. Kang, D. Shao, and J. Wang, "A fair electronic payment system for digital content using elliptic curve cryptography," *Journal of Algorithms & Computational Technology*, vol. 12, no. 1, pp. 13–19, 2018.

[7] B. Y. Kang, M. Wang, and D. Y. Jing, "An off-line payment scheme for digital content via subliminal channel," *Journal of Information Science & Engineering*, vol. 34, no. 1, 2018.

[8] B. Kang, J. Wang, and D. Shao, "Certificateless public auditing with privacy preserving for cloud-assisted wireless body area networks," *Mobile Information Systems*, vol. 2017, pp. 5, 2017.

[9] M. Kumar, C. P. Katti, and P. C. Saxena, "An untraceable identity-based blind signature scheme without pairing for E-cash payment system," in *International Conference on Ubiquitous Communications and Network Computing*, pp. 67–78, 2017.

[10] S. Li, X. Hu, Y. Zhang, W. Dong, J. Ye, H. Sun, *et al.*, "Research on offline transaction model in mobile payment system," in *International Conference on Frontier Computing*, pp. 1815–1820, 2018.

[11] Y. Li, F. Zhou, and Z. Xu, "A fair offline electronic cash scheme with multiple-bank in standard model," *Journal of the Chinese Institute of Engineers*, vol. 42, no. 1, pp. 87–96, 2019.

[12] B. Lian, G. Chen, J. Cui, and D. He, "Compact E-cash with practical and complete tracing," *KSII Transactions on Internet & Information Systems*, vol. 13, no. 7, 2019.

[13] H. Liu, Y. Chen, H. Tian, and T. Wang, "A secure and efficient data aggregation scheme for cloud-assisted wireless body area network," *International Journal of Network Security*, vol. 21, no. 2, pp. 243-249, 2019.

[14] J. N. Luo and M. H. Yang, "Offline transferable E-cash mechanism," in *IEEE Conference on Dependable and Secure Computing (DSC'18)*, pp. 1–2, 2018.

[15] J. Muleravicius, I. Timofejeva, A. Mihalkovich, and E. Sakalauskas, "Security, trustworthiness and effectivity analysis of an offline E-cash system with observers," *Informatica*, vol. 30, no. 2, pp. 327–348, 2019.

[16] S. Preibusch, T. Peetz, G. Acar, and B. Berendt, "Shopping for privacy: Purchase details leaked to paypal," *Electronic Commerce Research and Applications*, vol. 15, pp. 52–64, 2016.

[17] S. Rezaei, M. A. Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115–131, 2018.

[18] S. Sung, E. Kong, and C. Youn, "Mobile payment based on transaction certificate using cloud self-proxy server," *Etri Journal*, vol. 39, no. 1, pp. 135–144, 2017.

[19] P. Y. Ting, J. L. Tsai, and T. S. Wu, "Signcryption method suitable for low-power LoT devices in a wireless sensor network," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2385–2394, 2017.

[20] R. Tso and C. Y. Lin, "An off-line mobile payment protocol providing double-spending detection," in *The 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA'17)*, pp. 570–575, 2017.

[21] L. Wang, J. Gao, and X. Li, "Efficient bitcoin password-protected wallet scheme with key-dependent message security," *International Journal of Network Security*, vol. 21, no. 5, pp. 774–784, 2019.

[22] C. H. Wei, M. S. Hwang, and A. Y. H. Chin, "An improved authentication protocol for mobile agent device in RFID environment," *International Journal of Mobile Communications*, vol. 10, no. 5, pp. 508–520, 2012.

[23] J. H. Yang and P. Y. Lin, "A mobile payment mechanism with anonymity for cloud computing," *Journal of Systems and Software*, vol. 116, pp. 69–74, 2016.

[24] F. Zamanian and H. Mala, "A new anonymous unlinkable mobile payment protocol," in *The 6th International Conference on Computer and Knowledge Engineering (ICCKE'16)*, pp. 117–122, 2016.

Biography

Baoyuan Kang received M.S. in algebra from the Shanxi University, and ph.D. in cryptography from Xidian University, People's Republic of China in 1993 and 1999, respectively. He is a professor at Tianjin Polytechnic University. His current research interests are cryptography and information security.

Jianqi Du received the B.S. degree in internet of things engineering from Qiqihar University, China, in 2017. He is currently pursuing his M.S. degree at Tianjin Polytechnic University. His research interests are cryptography and information security.

Yanbao Han received the B.S. degree in machine design and manufacturing and its automation from Xiamen University of Technology, China, in 2015. He is currently pursuing his M.S. degree at Tianjin Polytechnic University. His research interests are cryptography and information security.

Kun Qian received the B.S. degree in computer science and technology from Tianjin University, China, in 2018. He is currently pursuing his M.S. degree at Tianjin Polytechnic University. His research interests are cryptography and information security.

Experimental Study on the Influence of Satellite Spoofing on Power Timing Synchronization

Jianwu Zhang[1], Xinyu Luo[1], Xingbing Fu[2], Xuxu Wang[1], Chunsheng Guo[1], and Yanan Bai[3]
(Corresponding author: Xingbing Fu)

School of Communication Engineering, Hangzhou Dianzi University[1]
Hangzhou, Zhejiang Province, China
School of Cyberspace, Hangzhou Dianzi University[2]
Hangzhou, Zhejiang Province, China
Chongqing Key Laboratory of Automated Reasoning and Cognition,
Chongqing Institute of Green and Intelligent Technology[3]
Chinese Academy Sciences, Chongqing, China
(Email: fuxbuestc@126.com)
(Received July 3, 2019; Revised and Accepted Dec. 3, 2019; First Online Feb. 3, 2020)

Abstract

The accuracy of time source in power system is required to be high, and the time benchmark of ground usually synchronizes with satellite time. The experimental environment is operated in the open outdoor area. SMBV100A is used as the pseudo Beidou signal transmitter and ATGM332D-5N is used as the Beidou signal receiving chip. The experiment changes the power of pseudo-Beidou satellite signal to explore the influence of spoofing jamming on satellite-ground time synchronization of a single receiving module, and changes the relative position of two receiving modules to study the influence of spoofing jamming on satellite common-view time synchronization. The experimental results show that in the case of deceptive jamming, the greater the power of pseudo-Beidou signal, the shorter the time interval between the receiving module and pseudo-Beidou; when the dual receiving module uses satellite common-view synchronization, there is a large time difference in a period of time. Experiments directly show that time-service time in power system is susceptible to spoofing interference

Keywords: Beidou Spoofing Jamming; Satellite Common View; Time Synchronization between Satellite and Ground; Time Synchronization in Smart Grid

1 Introduction

In the power system, each power automation device, microcomputer monitoring device, and safety automatic protection device are highly dependent on clock synchronization [7, 12]. In order to ensure the accuracy of clock synchronization, the timing clock in the power system is mostly synchronized with the satellite clock by means of satellite synchronization [11,19]. However, due to the navigation message format of the civilian part of the satellite signal including Beidou and GPS, the code modulation mode, carrier frequency and other information are all public. It is easy to use this information to design the Beidou satellite simulator to deceive the satellite receiver,which makes the timing clock of the ground having an error [2, 15, 17].

There are two main ways of time synchronization in the power system:master-slave time synchronization and satellite common-view time synchronization [9, 18]. Master-slave clock synchronization relies on the accuracy of single receiver satellite time synchronization. Satellite common-view time synchronization relies on time synchronization accuracy between multiple satellite receivers [5, 6, 10]. In the case of deception jamming, this work studies the change of the satellite synchronization time of the single Beidou receiver and the change of the satellite synchronization time synchronization of the dual receiver.

2 Experimental Principle

2.1 Satellite-Ground Time Synchronization

Satellites in space continuously transmit satellite signals to the ground by broadcasting, and the ground receiver can resolve the position of the satellite and the time stamp of transmitting signal from the satellite signals.

In theory, when four satellites are used, the ground time and satellite time can be synchronized by Formu-

las (1), (2), (3), and (4) [13].

$$\begin{aligned}
\rho_1 &= c(t_1 - t) && (1)\\
&= \sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2} + cb\\
\rho_2 &= c(t_2 - t) && (2)\\
&= \sqrt{(x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2} + cb\\
\rho_3 &= c(t_3 - t) && (3)\\
&= \sqrt{(x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2} + cb\\
\rho_4 &= c(t_4 - t) && (4)\\
&= \sqrt{(x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2} + cb.
\end{aligned}$$

In Formulas (1), (2), (3) and (4), the subscripts 1, 2, 3 and 4 represent the satellite signal sequence number, ρ_i is the pseudo range of the transmitting satellite to the receiver, t_i is the time stamp of the transmission, and t is local time. c is the speed of light , (x, y, z) is the local location, (x_i, y_i, z_i) is the location where the satellite is launched, and b is the deviation of local time from standard satellite time. It is known that (x_i, y_i, z_i) , t_i and t can find four unknown quantities x , y , z and b , and find $(t - b)$ is the synchronization time between the receiver and the satellite.

2.2 Deception Interference Time Synchronization Principle

When the ground satellite receiver performs the satellite time synchronization normally, it will first capture the signals of different satellite serial numbers from the received signals, and then keep track of the satellite signals. However, the power of the satellite signal received by the ground receiver is relatively small, only about -160dBW [8, 14].

If a pseudo-satellite signal with a relatively large power is transmitted near the receiving module, the pseudo-satellite signal will mask the real satellite signal. The receiving module thereby captures the tracking pseudo lite signal and synchronizes the pseudo lite time according to the navigation message of the pseudo lite signal [3, 4, 16].

2.3 Satellite Common-view Time Synchronization Principle

Satellite common view is a method of time synchronization between devices based on satellite time. Figure 1 shows the schematic diagram of satellite common-view time synchronization.

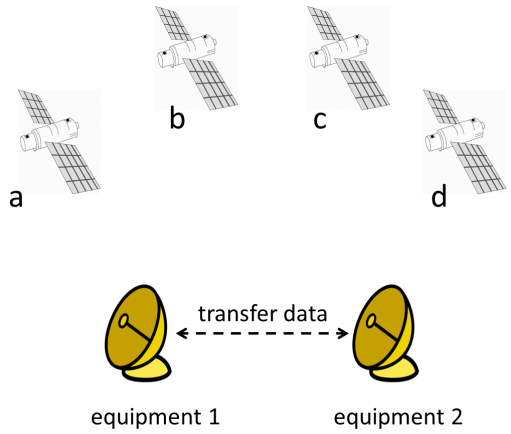


Figure 1: Satellite common view

In Figure 1, device A and device B simultaneously receive satellite signals for satellite time synchronization. Since the durations of the signal processing and data transmitting of device A are different from those of device B, there is a relatively fixed time offset between A and B [1].

3 Experimental Equipment and Experimental Environment

3.1 Experimental Equipment

The main equipment of this experiment is GPS/BD dual-mode receiver module and pseudo-Beidou signal simulator equipment. The GPS/BD dual-mode receiving module includes three parts: Receiving antenna, signal processing chip and serial interface tool. The GPS/BD dual-mode receiving module is shown in Figure 2.

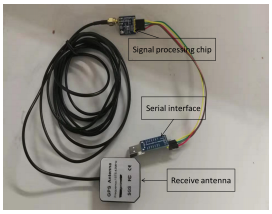


Figure 2: GPS/BD receiving module

The receiving antenna is an ordinary GPS/BD antenna for receiving signals; The signal processing chip is ATGM332D-5N, which is used to capture, track, and analyze satellite signals, and calculates local position and time information; The serial interface tool is CP 2102 USB- TTL BOARDV4.0, used to transfer chip processing data to a computer. Among them, ATGM332D-5N is a key part of the receiving module. Its positioning accuracy is 2.5m, cold-start capture accuracy is -148dBm and tracking capture sensitivity is -162dBm.

The pseudo-Beidou signal simulator is SMBV100A, which is a vector signal generator produced by ROHDE & SCHWARZ. It can simulate the transmission of 12 Beidou

satellite signals, and can set the power of pseudo-satellite signals, UTC, and the location of pseudo-satellite signals.

This experiment records the processing data of the receiving module through the serial port assistant on the computer.

3.2 Experimental Environment Construction

The system block diagram of this experiment is shown in Figure 3. The receiving module receives the real GPS/BD satellite signal during normal operation; when the receiving module is deceived, it receives the real GPS/BD signal and the pseudo-high-pitched signal with high power. The received satellite signal is processed by the serial port tool on the computer.

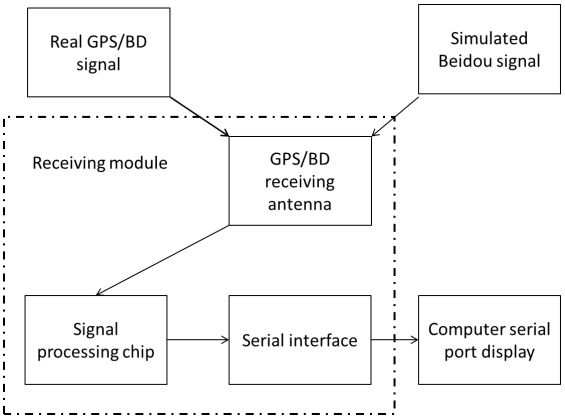


Figure 3: The block diagram of deception interference experiment system

This experiment is carried out in an outdoor open environment to ensure that the receiving module can receive more satellite signals. The transmitting antenna of the pseudo-Beidou signal simulator is placed on a higher floor, ensuring that the pseudo-Beidou signal can cover a larger area. The experimental scene of the BD/GPS receiving module is shown in Figure 4.

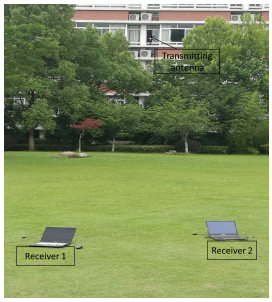


Figure 4: The experimental site map of BD/GPS receiver module

The pseudo-Beidou signal simulator SMBV100A (Figure 5) placed on the upper floor simulates the pseudo-Beidou signal transmitted to the receiving module by a non-omnidirectional antenna with a small beam angle.

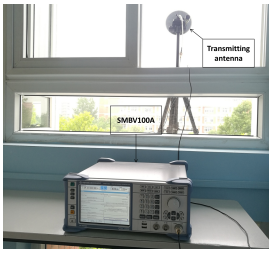


Figure 5: Pseudo-beidou signal simulator and signal transmitting antenna

4 Experimental Steps

This experiment separately studies the effect of deception jamming on the time synchronization of the single receiver module and the effect of deception jamming on the satellite synchronization time synchronization of the dual receiver module. The experimental steps are as follows:

1) Turn off the signal transmission switch of the pseudo-Beidou signal simulator, the receiving module normally receives the real GPS, BD satellite signal and records the signal processing data for 5 minutes through the serial port assistant of the computer after the data is stable.

2) Set the transmission power of the pseudo signal to -20dBm; set the pseudo positioning position of the Beidou signal to be (30°18'53"N, 120°22'23"E), which is about 2' difference from the longitude of the real position; The UTC of the interference source is set to 08:00:00, which is about 12 hours from the real time; the pseudo satellite signal number is 01, 02, 03, 04, 05, 06. Turn on the pseudo-Beidou signal simulator signal emission switch.

3) The serial port assistant records 5 minutes of signal processing data. Turn off the signal transmission switch of the pseudo-Beidou signal simulator.

4) Study the effect of deception jamming on the time synchronization of the single receiver module. Ensure that the experimental environment is unchanged, set the UTC to 04:00:00, change the power of the pseudo-Beidou signal to -20dBm, -15dBm, -10dBm and -5dBm, and repeat the experimental steps from a to c.

5) The effect of deception jamming on the dual-receiving module on satellite common-view time synchronization is studied. Ensure that the experimental environment is unchanged, set the UTC to 07:00:00, the power of the pseudo-Beidou signal to -5dBm, and change the relative distance between the dual receivers to 5m, 10m, 15m and 20m. Experimental steps from a to c were repeated in sequence.

5 Experimental Results and Analysis

5.1 Deception of the Receiving Module

The receiving module normally receives the BD/GPS data, and the GNRMC data read by the computer serial port is shown in Figure 6.

```
$GPGSV,3,3,09,30,19,089,25*4D
$BDGSV,3,1,10,02,,,20,03,,,29,04,,,23,05,,,22*63
$BDGSV,3,2,10,06,72,232,26,08,78,098,30,09,47,224,27,11,21,158,19*65
$BDGSV,3,3,10,13,75,337,29,14,45,036,34*63
$GNRMC,104937.000,A,3018.8702,N,12020.3913,E,0.00,20.86,170718,,,A*42
$GNVTG,20.86,T,,M,0.00,N,0.00,K,A*1F
$GNZDA,104937.000,17,07,2018,00,00*4A
$GPTXT,01,01,01,ANTENNA OK*35
$GNGGA,104938.000,3018.8702,N,12020.3913,E,1,14,0.9,29.0,M,0.0,M,,*4C
$GNGLL,3018.8702,N,12020.3913,E,104938.000,A,A*4E
$GPGSA,A,3,02,06,30,13,05,15,07,29,,,,,1.5,0.9,1.2*34
$BDGSA,A,3,06,08,13,14,11,09,,,,,,,1.5,0.9,1.2*2D
$GPGSV,3,1,09,02,70,083,40,05,61,353,25,06,24,115,32,07,12,060,24*78
$GPGSV,3,2,09,13,59,184,43,15,25,216,20,19,05,161,,29,34,308,32*7C
$GPGSV,3,3,09,30,19,089,25*4D
$BDGSV,3,1,10,02,,,20,03,,,29,04,,,23,05,,,25*64
$BDGSV,3,2,10,06,72,232,26,08,78,098,31,09,47,224,27,11,21,158,19*64
$BDGSV,3,3,10,13,75,337,31,14,45,036,34*6A
$GNRMC,104938.000,A,3018.8702,N,12020.3913,E,0.00,20.86,170718,,,A*4D
$GNVTG,20.86,T,,M,0.00,N,0.00,K,A*1F
```

Figure 6: Serial GNRMC data under normal conditions

The serial port results show that the UTC is 10:49:37 and the local location is (3018.8702N, 12020.3913E), which is (30°18'87.02"N, 120°20'39.13"E). The synchronization time and the positioning position are consistent with the local time and local location, indicating that the receiving module has not received fraudulent interference.

After about 5 minutes of transmitting the pseudo Beidou signal, the GNRMC data read by the serial port is shown in Figure 7. The result shows that the UTC is 08:56:33, and the positioning position is (3018.8835N, 12022.3833E), which is consistent with the parameters set by the Beidou simulator, and is inconsistent with the local time and position. The receiving module is deceived.

```
$BDGSV,3,2,10,05,31,242,54,06,44,266,53,07,26,170,53,08,25,285,52*6C
$BDGSV,3,3,10,13,30,285,53,14,34,223,53*66
$GNRMC,085633.000,A,3018.8835,N,12022.3833,E,0.00,20.86,160718,,,A*4A
$GNVTG,20.86,T,,M,0.00,N,0.00,K,A*1F
$GNZDA,085633.000,16,07,2018,00,00*48
$GPTXT,01,01,01,ANTENNA OK*35
$GNGGA,085634.000,3018.8835,N,12022.3833,E,1,10,1.3,-0.7,M,0.0,M,,*53
$GNGLL,3018.8835,N,12022.3833,E,085634.000,A,A*4F
$GPGSA,A,3,,,,,,,,,,,,,2.7,1.3,2.4*33
$BDGSA,A,3,03,07,05,04,13,06,08,01,02,14,,,2.7,1.3,2.4*2D
$GPGSV,3,1,12,02,50,328,35,04,,,26,05,46,252,16,06,52,051,*40
$GPGSV,3,2,12,09,33,059,26,12,29,267,,13,06,190,36,15,00,212,28*77
$GPGSV,3,3,12,17,33,145,27,19,57,142,28,23,08,039,29,25,11,308,29*71
$BDGSV,3,1,10,01,36,123,54,02,47,219,54,03,54,164,54,04,19,108,53*67
$BDGSV,3,2,10,05,31,242,54,06,44,266,53,07,26,170,53,08,25,285,52*6C
$BDGSV,3,3,10,13,30,285,53,14,34,223,53*66
$GNRMC,085634.000,A,3018.8835,N,12022.3833,E,0.00,20.86,160718,,,A*4D
$GNVTG,20.86,T,,M,0.00,N,0.00,K,A*1F
$GNZDA,085634.000,16,07,2018,00,00*4F
$GPTXT,01,01,01,ANTENNA OK*35
```

Figure 7: Serial GNRMC data in case of fraudulent interference

5.2 Satellite-Ground Time Synchronization under Deception

When the receiving module receives the real Beidou signal, the transmission power of the pseudo Beidou signal is changed to -20dBm, -15dBm, -10dBm and -5dBm, and the experimental data simulation results are shown in Figures 8–11.

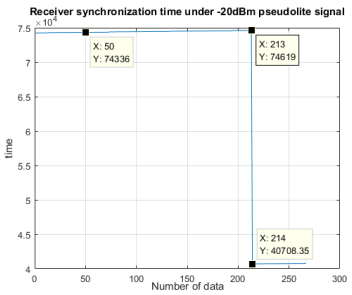


Figure 8: Receiver synchronization time under -20dBm pseudo lite signal

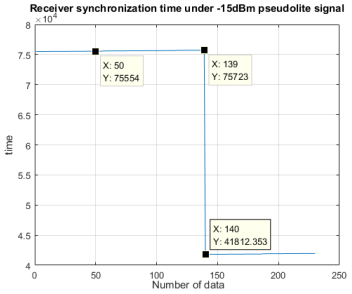


Figure 9: Receiver synchronization time under -15dBm pseudo lite signal

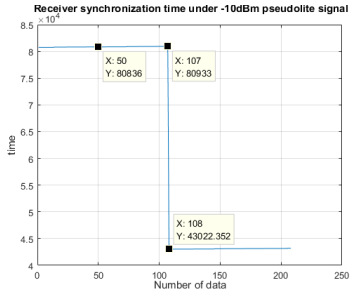


Figure 10: Receiver synchronization time under -10dBm pseudo lite signal

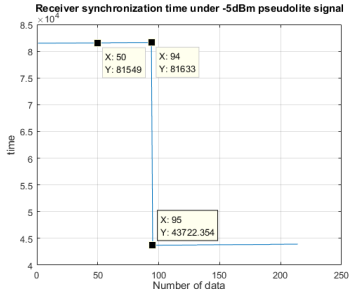


Figure 11: Receiver synchronization time under -5dBm pseudo lite signal

In Figures 8–11, the abscissa is the number of data frames received by the serial port, and 1 s corresponds to one data point. The ordinate is the UTC resolved by the receiving module and is represented by scientific notation.

In Figures 8–11, the UTC line starts from the left and the first black point indicates the moment when

the pseudo-beidou signal is transmitted, the second black point indicates the last moment of the synchronized real satellite time, and the third black point indicates the first moment of time of the synchronous pseudo-beidou. It can be seen that after the pseudo-Beidou signal is transmitted, the UTC parsed by the receiving module does not immediately synchronize the pseudo-dipper time, and still maintains the real satellite time before the interference.

The time interval between the first black point and the second black point indicates the time required for synchronizing the satellite time and the pseudo-beidou signal in the case of fraudulen interference. According to Figures 8–11, the length of time required for synchronizing the UTC of the receiving module and the pseudo Beidou signal is recorded in the case of pseudo-Beidou signal fraud interference of different powers, as shown in Table 1.

Table 1: Time required for synchronizing pseudo-beidou signals

Pseudo-Beidou Signal Power/dBm	-20	-15	-10	-5
Synchronized Pseudo-beidou Signal Time/s	163	89	57	44

It can be seen from Table 1 that the greater the power of the pseudo-Beidou signal, the shorter the time that the UTC resolved by the receiving module is synchronized to the pseudo-Beidou signal.

5.3 Satellite Common Time Synchronization under Deception Jamming

The power of the pseudo-Beidou signal is kept at -5dBm, and the relative distance between the two receivers is changed, which is 5m, 10m, 15m and 20m, respectively, and the experimental data simulation results are obtained (Figures 12–15). In Figures 12–15, it is ensured that the two receiver modules have the same abscissa of the UTC at the same moment. The significance of the three black points on the two UTC lines in the figure is still the pseudo-Beidou signal transmitting time, the last moment of the synchronized real satellite time, and the first moment of time of the synchronous pseudo-beidou is analyzed.

In Figures 12–14, both the receivers 1 and 2 have a UTC hopping condition, and in Figure 15, the receiver 1 always maintains the true UTC, indicating that it is not subject to spoofing interference.

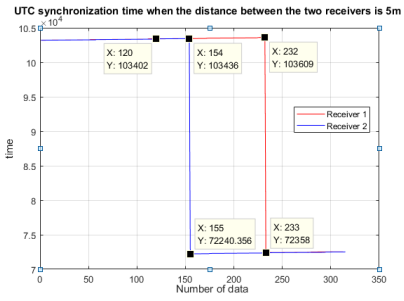


Figure 12: UTC synchronization time when the distance between the two receivers is 5m

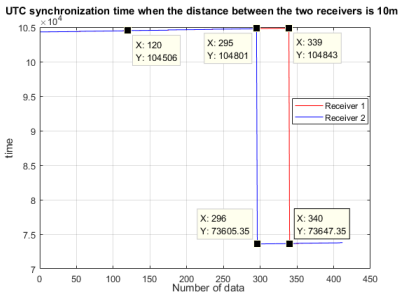


Figure 13: UTC synchronization time when the distance between the two receivers is 10m

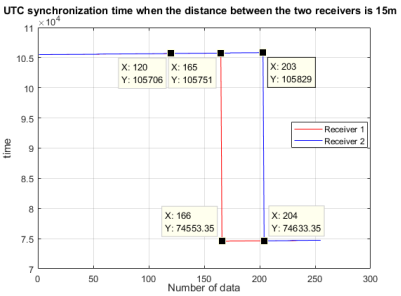


Figure 14: UTC synchronization time when the distance between the two receivers is 15m

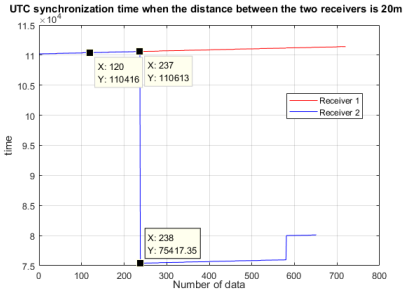


Figure 15: UTC synchronization time when the distance between the two receivers is 20m

As can be seen from Figure 12 to Figure 15, in the case of fraudulent interference, the dual-receiving module satellite common-view synchronization time has a large time difference over a period of time. Record the length of time when the synchronization time of the two receivers

is different under different relative distances. The results are shown in Table 2.

Table 2: Dual receiver satellite common-view synchronous UTC deviation duration

Relative Distance Between Two Receivers/m	5	10	15	20
Satellite Common-view Time Deviation Duration/s	78	44	38	–

It can be seen from Table 2 that when the relative distance between the receivers is too large, the time deviation between the two receiving modules is always large because the pseudo-Beidou signal cannot cause deception to interfere with one of the receiving modules. When the relative distance between the receiving modules is less than 20m, the time deviation between the two receiving modules will be relatively large during a period of time, about 40s to 80s.

6 Conclusion

In this work, based on two time synchronization methods of timing time source in the power system: Satellite-Ground Time Synchronization and Satellite Common Time Synchronization, experiment changes the transmission power of the pseudo-Beidou time signal and the distance between the two receivers to explore the satellite fraudulent interference. The following conclusions are obtained:

1) After being deceived, the time of synchronization of the satellite will jump to the pseudo-Beidou signal time.

2) The greater the power of the pseudo Beidou signal, the shorter the time that synchronizes the receiving module and the pseudo Beidou signal.

3) When the distance between two receivers based on satellite common view synchronization is too large, the pseudo-Beidou signal transmitted by the Beidou simulator may not be able to successfully deceive one of the receivers. The time offset between the two receivers is kept large.

4) When the relative distance of the receiver is within 20m, the dual receivers based on the satellite common view synchronization are deceptively interfered, and the synchronization time will have a large deviation during a period of time, about 40s to 80s.

Acknowledgments

The authors would like to thank the Joint Funds of Smart Grid of the National Natural Science Foundation of China (No.U1866209) for providing the project foundation.

References

[1] D. W. Allan and M. A. Weiss, "Accurate time and frequency transfer during common-view of a GPS satellite," in *The 34th Annual Symposium on Frequency Control*, pp. 334–346, May 1980.

[2] C. Bonebrake and L. Ross O'Neil, "Attacks on GPS time reliability," *IEEE Security Privacy*, vol. 12, pp. 82–84, 2014.

[3] S. Daneshmand, A. Jafarnia-Jahromi, A. Broumandan, and G. Lachapelle, "A low-complexity GNSS spoofing mitigation technique using a double antenna array," *GPS World Magazine*, vol. 22, pp. 44–46, 2011.

[4] S. Daneshmand, A. Jafarnia-Jahromi, A. Broumandan, and G. Lachapelle, "A low-complexity GPS anti-spoofing method using a multi-antenna array," in *Proceedings of the 25th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS'12)*, pp. 1233–1243, Sep. 2012.

[5] Y. J. Huang, J. D. Fu, H. Takiguchi, W. H. Tseng, and H. W. Tsao, "Stability improvement of an operational two-way satellite time and frequency transfer system," *Metrologia*, vol. 53, pp. 881–890, 2016.

[6] J. F. Li, H. Li, C. X. Peng, J. Wen, and M. Lu, "Research on the random traversal raim method for anti-spoofing applications," in *China Satellite Navigation Conference (CSNC'19)*, pp. 593-605, May 2019.

[7] B. F. Liu, Z. Q. Ji, and Z. X. Xie, "The application analysis of IEEE 1588 in smart substation," in *The International Conference on Advanced Power System Automation and Protection (APAP'11)*, pp. 309–395, Oct. 2011.

[8] B. F. Liu, Z. Q. Ji, and Z. X. Xie, "GPS anti-spoofing technology based on RELAX algorithm in smart grid," in *Proceedings of the 10th International Conference on Communications and Networking in China Chinacom*, pp. 637–642, Aug. 2015.

[9] H. P. Liu, F. Yang, J. Zhou, and H. S. Zhao, "Application of time synchronization technology of beidou navigation system in power system," *East China Electric Power (in Chinese)*, vol. 39, pp. 489–491, 2011.

[10] Z. Y. Chen, H. Li; M. Q. Lu, "GNSS spoofing detection with single moving antenna based on the correlation of satellite transmit time residual," in *China Satellite Navigation Conference (CSNC'18)*, pp. 978–981, May 2018.

[11] B. Moussa, M. Debbabi, and C. Assi, "Security assessment of time synchronization mechanisms for the smart grid," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 1952–1973, 2016.

[12] F. Ramos, J. L. Gutierrez-Rivas, J. Lopez-Jimenez, B. Caracuel, and J. Diaz, "Accurate timing networks for dependable smart grid applications," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 2076–2084, 2018.

[13] N. O. Tippenhauer, K. B. Rasmussen, and S. Capkun, "On the requirements for successful gps spoofing attacks," in *ACM Conference on Computer and Communications Security*, pp. 75–86, Oct 2011.
[14] F. Wang, H. Li, and M. Lu, "GNSS spoofing countermeasure with a single rotating antenna," *Journals & Magazines*, vol. 5, pp. 8039–8047, 2017.
[15] K. Wesson, M. Rothlisberger, and T. Humphreys, "Practical cryptographic civil GPS signal authentication," *Navigation-Journal of The Institute of Navigation*, vol. 59, pp. 177–193, 2012.
[16] Y. T. Zhang, L. Wang, W. Y. Wang, D. Lu, and R. B. Wu, "Spoofing jamming suppression techniques for GPS based on DOA estimating," in *China Satellite Navigation Conference (CSNC'14)*, pp. 683–693, May 2014.
[17] Z. Zhang, S. Gong, A. D. Dimitrovski, and H. Li, "Time synchronization attack in smart grid: Impact and analysis," *IEEE Transactions on Smart Grid*, vol. 4, pp. 87–98, 2013.
[18] S. Zhao, "Measurement and synchronization technology based on satellite common view method," *Computer Measurement and Control (in Chinese)*, vol. 12, pp. 49–52, 2016.
[19] T. Zhao, Z. W. Li, and B. Zou, "Wide-area time synchronization method for satellite clock and network clock equipment," *Automation of Electric Power System (in Chinese)*, vol. 14, pp. 202–207, 2017.

Biography

Jianwu Zhang is a professor at Hangzhou Dianzi University, and he received a Ph.D. from Zhejiang University in 1999. His research interests include mobile communication and image processing.

Xinyu Luo is currently pursuing his master's degreee in electronics and communication engineering, Hangzhou Dianzi University. Her research interests include Beidou fraud detection, and image processing.

Xingbing Fu is a lecturer, and he received the Ph.D. degree from University of Electronic Science and Technology of China (UESTC) in 2016. His research interests include cloud computing, cryptography and information security.

Xuxu Wang is currently pursuing his master's degreee in electronics and communication engineering, Hangzhou Dianzi University. His research interests Beidou fraud detection.

Chunsheng Guo received a B.Sc.in Radio Engineering from the Northeastern University (NEU), Shenyang, China, and received a Ph.D. in Communication and Information System from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1993 and 2002 respectively. He is currently an associate professor at the School of Communication Engineering, Hangzhou Dianzi University, China. His research involves image segmentation, video moving objects detection, Video action recognition and Video anomaly detection.

Yanan Bai is a Ph.D. candidate from Chongqing Institute of Green and Intelligent Technology, Chinese Academy Sciences. Her research interests are homomorphic encryption and applications, big data privacy protection and Cryptography theory.

Some Further Results of Pseudorandom Binary Sequences Derived from the Discrete Logarithm in Finite Fields

Vladimir Edemskiy[1], Zhixiong Chen[2], and Sergey Garbar[3]
(Corresponding author: Vladimir Edemskiy)

Department of Applied Mathematics and Informatics, Novgorod State University[1,3]
Veliky Novgorod, 173003, Russia
Provincial Key Laboratory of Applied Mathematics, Putian University[2]
Putian, Fujian 351100, P. R. China
(Email: vladimir.edemskiy@novsu.ru)
(Received Feb. 13. 2020; revised and accepted June 8, 2020)

Abstract

In this paper, we study the linear complexity of pseudorandom binary sequences of period p^r (an odd prime power) derived from the discrete logarithm in finite fields. We determine the exact values of the linear complexity of the sequences for odd $r \geq 3$ and the k-error linear complexity for $k < (p-1)/2$. The results extend that of Z. Chen and Q. Wang from an earlier work.

Keywords: Binary Sequences; Finite Field; k-Error Linear Complexity

1 Introduction

Legendre sequence introduced below is a classic sequence defined using (multiplicative) character of finite fields. They have been extensively studied in the literature. The Legendre sequences have a lot of interesting randomness properties, in particular, they are quite good from the linear complexity viewpoint.

Let p be an odd prime, the *Legendre sequence* $\ell = (\ell_0, \ell_1, \ldots, \ell_{p-1})$ of period p is defined as

$$\ell_n = \begin{cases} 0, & \text{if } n = 0, \\ \frac{1-\left(\frac{n}{p}\right)}{2}, & \text{if, } 1 \leq n < p, \end{cases} \qquad \ell_{n+p} = \ell_n,$$

where $\left(\frac{\cdot}{p}\right)$ is the Legendre symbol. Legendre sequence has a number of good properties, the reader is referred to [4–6, 8] for details.

Very recently, Z. Chen and Q. Wang [2] discussed a generalization of Legendre sequence over the extension field $\mathbb{F}_q$ of $\mathbb{F}_p$, where $q = p^r$. Let $\{\gamma_1 = 1, \gamma_2, \ldots, \gamma_r\}$ be a fixed basis of $\mathbb{F}_q$ over $\mathbb{F}_p$, then define the element of $\mathbb{F}_q$ as follows

$$\xi_n = n_1\gamma_1 + n_2\gamma_2 + \cdots + n_r\gamma_r,$$

if

$$\begin{aligned} n &= n_1 + n_2 p + \cdots + n_r p^{r-1}, \\ &\qquad 0 \leq n < q, \quad 0 \leq n_i < p, \\ i &= 1, \ldots, r. \end{aligned}$$

Then the binary sequence $\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_{q-1})$ is defined by

$$\sigma_n = \frac{1 - \chi(\xi_n)}{2}, \quad 0 \leq n < q, \tag{1}$$

where χ is the quadratic character of $\mathbb{F}_q$ and we add $\chi(0) = 1$ (here and hereafter). We can set $\sigma_{n+q} = \sigma_n$ to get a periodic sequence. Clearly, we have $\chi(\xi_n) = (-1)^{\sigma_n}$. The σ had indeed been investigated in earlier references [9–11, 13]. It should be noted that these sequences are not cyclotomic sequences for $r > 1$. They have a different structure, and our method differs from the methods used in [1, 7, 14, 15] to study the linear complexity and k- error linear complexity of the cyclotomic sequences with period p^r.

Z. Chen and Q. Wang [2] proved a lower bound on the linear complexity of σ for $r \geq 2$ and determined the exact values of the k-error linear complexity for $r = 2$. In this work, we continue this project. Exactly speaking, we will present the exact values of the linear complexity of σ for odd $r \geq 3$ in Sect. 3 and prove some partial results about the k-error linear complexity for $r \geq 3$ in Sect. 4. Some subsidiary statements will be presented in Sect. 2. Conclusions will be drawn in Sect. 5, some final remarks are also presented there.

Finally we conclude this section by introducing the notions of the linear complexity and the k-error linear complexity. Let $\mathbb{F}$ be a field. For a T-periodic sequence (s_n) over $\mathbb{F}$, recall that the *linear complexity* over $\mathbb{F}$, denoted by $LC^{\mathbb{F}}((s_n))$, is the least order L of a linear recurrence

relation over $\mathbb{F}$

$$s_{n+L} = c_{L-1}s_{n+L-1} + \ldots + c_1 s_{n+1} + c_0 s_n \quad \text{for } n \geq 0,$$

which is satisfied by (s_n) and where $c_0 \neq 0, c_1, \ldots, c_{L-1} \in \mathbb{F}$. Let

$$S(X) = s_0 + s_1 X + s_2 X^2 + \ldots + s_{T-1} X^{T-1} \in \mathbb{F}[X],$$

which is called the *generating polynomial* of (s_n). Then the linear complexity over $\mathbb{F}$ of (s_n) can be computed as

$$LC^{\mathbb{F}}((s_n)) = T - \deg\left(\gcd(X^T - 1,\ S(X))\right), \tag{2}$$

which is the degree of the *characteristic polynomial*, $(X^T - 1)/\gcd(X^T - 1,\ S(X))$, of the sequence. See, *e.g.*, [3] for details.

A cryptographically strong sequence must have high linear complexity. At the same time, changing several members of such a sequence should not significantly reduce the linear complexity. This leads to the concept of the k-error linear complexity. For integers $k \geq 0$, the *k-error linear complexity* over $\mathbb{F}$ of (s_n), denoted by $LC_k^{\mathbb{F}}((s_n))$, is the lowest linear complexity (over $\mathbb{F}$) that can be obtained by changing at most k terms of the sequence per period (see [12], and see [6] for the related sphere complexity that was defined even earlier). Clearly, $LC_0^{\mathbb{F}}((s_n)) = LC^{\mathbb{F}}((s_n))$, and

$$T \geq LC_0^{\mathbb{F}}((s_n)) \geq LC_1^{\mathbb{F}}((s_n)) \geq \ldots \geq LC_w^{\mathbb{F}}((s_n)) = 0$$

when w equals the number of nonzero terms of (s_n) per period, *i.e.*, the weight of (s_n).

2 A Subsidiary Polynomial

For our discussion, we need to introduce a subsidiary polynomial.

For $1 \leq m \leq r$, we use $\gamma_1 = 1, \gamma_2, \ldots, \gamma_m$ (as in Sect. 1) to build the set $\mathbb{L}_m = \{i_1\gamma_1 + i_2\gamma_2 + \ldots + i_m\gamma_m : 0 \leq i_1, i_2, \ldots, i_m < p\}$. For $\beta \in \mathbb{L}_m$, we define

$$\phi^{(m)}(\beta) = \prod_{i_{m+1}=0}^{p-1} \cdots \prod_{i_r=0}^{p-1} \chi(\beta + i_{m+1}\gamma_{m+1} + \ldots + i_r\gamma_r).$$

Then we introduce the subsidiary polynomial

$$\mathcal{A}^{(m)}(X) = \sum_{n=0}^{p^m-1} a_n^{(m)} X^n \in \mathbb{F}_2[X]$$

of degree $< p^m$ with coefficients

$$a_n^{(m)} = (1 - \phi^{(m)}(\rho_n))/2, \tag{3}$$

where

$$\rho_n = n_1\gamma_1 + n_2\gamma_2 + \cdots + n_m\gamma_m,$$

for $0 \leq n < p^m$ and

$$n = n_1 + n_2 p + \cdots + n_m p^{m-1},\ \ 0 \leq n_i < p, \quad i = 1, \ldots, m.$$

We have the following statement.

Proposition 1. *With notations as above. For $1 \leq m \leq r$ with odd $r \geq 3$, we have*

$$wt(\mathcal{A}^{(m)}(X)) = (p^m - 1)/2 + \delta_m,$$

here and hereafter $wt(h(X))$ denotes the number of nonzero coefficients of the polynomial $h(X)$, and

$$\delta_m = \begin{cases} 1, & \text{if } p \equiv 3 (\mathrm{mod}\ 4) \text{ and } m \text{ is even,} \\ 0, & \text{otherwise.} \end{cases}$$

In order to prove Proposition 1, we need to discuss some lemmas. Let $0 < g < p$ be a primitive root modulo p. In the sequel we will use the following notations:

$$\widehat{a} \equiv a \pmod{p} \text{ and } 0 \leq \widehat{a} < p,$$

and

$$g * b = \widehat{gb_0} + \widehat{gb_1} \cdot p + \ldots + \widehat{gb_l} \cdot p^l$$

for $b = b_0 + b_1 p + \ldots + b_l p^l$, where $0 \leq b_0, b_1, \ldots, b_l < p$.

Lemma 1. *Let χ be the quadratic character of $\mathbb{F}_q$ with $q = p^r$ and odd $r \geq 3$ and $g : 0 < g < p$ a primitive root modulo p. With $m, \mathbb{L}_m$ and $\phi^{(m)}$ as above, we have*

(i). $\chi(g) = -1$.

(ii). $\phi^{(m)}(g\beta) = -\phi^{(m)}(\beta)$ for any $\beta \in \mathbb{L}_m$.

Proof.

(i). Let ξ be a primitive element of $\mathbb{F}_q^*$. We can write $g = \xi^{\frac{t(q-1)}{p-1}}$ for some integer t since the order of g modulo p is $p-1$. We need to show $\gcd(t, p-1) = 1$. Since otherwise, if $d = \gcd(t, p-1) > 1$, we see that

$$g^{(p-1)/d} = \xi^{\frac{t(q-1)}{d}} = (\xi^{q-1})^{t/d} = 1,$$

a contradiction. So we derive that t is odd and $\frac{q-1}{p-1} = 1 + p + \cdots + p^{r-1}$ is odd too when r is odd. Hence $\chi(g) = -1$.

(ii). This comes from the definition of $\phi^{(m)}$ and (i).

□

Lemma 2. *Let $l \geq 1$ and $\mathbb{Z}_{p^l}$ be the ring of integers modulo p^l. Let $g : 0 < g < p$ be a primitive root modulo p. Then there exists a subset M such that $\mathbb{Z}_{p^l} \setminus \{0\} = M \cup g*M$ and $M \cap g*M = \emptyset$, where $g*M = \{g*j : j \in M\}$.*

Proof. Put

$$M = \bigcup_{i=0}^{l-1} \bigcup_{k=0}^{(p-3)/2} \{\widehat{g^{2k}} \cdot p^i + a_{i+1}p^{i+1} + \ldots + a_{l-1}p^{l-1} : 0 \leq a_{i+1}, \ldots, a_{l-1} < p\}.$$

We see that M exactly contains $\frac{p-1}{2}(p^{l-1} + \cdots + p + 1) = (p^l - 1)/2$ integers, so does $g * M$. It is clear that $M \cap g * M = \emptyset$. Hence $M \cup g * M = \mathbb{Z}_{p^l} \setminus \{0\}$. □

Now for $1 \le m \le r$ and $0 \le j < p^{m-1}$, we write

$$U_j^{(m)}(X) = \begin{cases} \sum_{k=0}^{p-1} a_{j+kp^{m-1}}^{(m)} X^{j+kp^{m-1}}, & \text{if } m > 1, \\ 0, & \text{if } m = 1, \end{cases}$$

and

$$V^{(m)}(X) = \sum_{k=0}^{p-2} a_{\widehat{g^k} \cdot p^{m-1}}^{(m)} X^{\widehat{g^k} \cdot p^{m-1}}.$$

Indeed, $U_0^{(m)}(X) = V^{(m)}(X) + a_0^{(m)}$ for $m > 1$. Hence it is clear to see that

$$\mathcal{A}^{(m)}(X) = a_0^{(m)} + \sum_{j=1}^{p^{m-1}-1} U_j^{(m)}(X) + V^{(m)}(X). \quad (4)$$

Lemma 3. *With notations as above, we have*

(i). $wt(V^{(m)}(X)) = (p-1)/2$.

(ii). $a_0^{(m)} = (1 - (-1)^{(p^{r-m}-1)/2})/2$.

(iii). There is a set $J \subset \mathbb{Z}_{p^{m-1}}$ *such that* $\mathbb{Z}_{p^{m-1}} \setminus \{0\} = J \cup g * J$ *and* $J \cap g * J = \emptyset$ *for* $m > 1$. *And for any* $j \in J$, *we have*

$$wt\left(U_j^{(m)}(X)\right) + wt\left(U_{g*j}^{(m)}(X)\right) = p.$$

Proof.

(i). From Equation (3), if $a_{\widehat{g^k} \cdot p^{m-1}}^{(m)} = 1$ then $\phi_m(\widehat{g^k} \cdot \gamma_m) = -1$. Hence, by Lemma 1 (ii) we have $\phi_m(\widehat{g^{k+1}} \cdot \gamma_m) = 1$, from which we derive $a_{\widehat{g^{k+1}} \cdot p^{m-1}}^{(m)} = 0$ and vice verse. Thus, $wt(V^{(m)}(X)) = (p-1)/2$.

(ii). By Equation (3) again, we only need to compute

$$\phi^{(m)}(0) = \sum_{i_{m+1}=0}^{p-1} \cdots \sum_{i_r=0}^{p-1} \chi(i_{m+1}\gamma_{m+1} + \ldots + i_r\gamma_r).$$

Let $\mathbb{M}_{r-m} = \{i_{m+1}\gamma_{m+1} + \ldots + i_r\gamma_r : 0 \le i_{m+1}, \ldots, i_r < p\}$. For any $\beta \in \mathbb{M}_{r-m}$, we have $g\beta \in \mathbb{M}_{r-m}$. Since $\chi(g\beta) = -\chi(\beta)$ for $\beta \ne 0$ by Lemma 1, we define two subsets of $\mathbb{M}_{r-m}$:

$$\{\beta : \chi(\beta) = 1, 0 \ne \beta \in \mathbb{M}_{r-m}\}$$

and

$$\{g\beta : \chi(\beta) = 1, 0 \ne \beta \in \mathbb{M}_{r-m}\},$$

both with the same cardinality $(p^{r-m}-1)/2$. So we get $\phi^{(m)}(0) = (-1)^{(p^{r-m}-1)/2}$.

(iii). By Lemma 2, such $J \subset \mathbb{Z}_{p^{m-1}}$ always exists.

Let $j = j_1 + j_2 p + \ldots + j_{m-1}p^{m-2} \in J$, where $0 \le j_1, j_2, \ldots, j_{m-1} < p$. For those $k : 0 \le k < p$ such that $a_{j+kp^{m-1}}^{(m)} = 1$, we have $\phi^{(m)}(j_1\gamma_1 + j_2\gamma_2 + \ldots + j_{m-1}\gamma_{m-1} + k\gamma_m) = -1$ and hence $\phi^{(m)}(\widehat{gj_1}\gamma_1 + \widehat{gj_2}\gamma_2 + \ldots + \widehat{gj_{m-1}}\gamma_{m-1} + \widehat{gk}\gamma_m) = 1$, which derives $a_{g*j+\widehat{gk}p^{m-1}}^{(m)} = 0$.

Similar, for those $k : 0 \le k < p$ such that $a_{j+kp^{m-1}}^{(m)} = 0$, we derive $a_{g*j+\widehat{gk}p^{m-1}}^{(m)} = 1$. So we complete the proof. □

Proof of Proposition 1. It follows from Equation (4) and Lemma 3. □

3 Linear Complexity

In this section, we present the main result on the linear complexity. We first describe a connection between the subsidiary polynomial $\mathcal{A}^{(m)}(X)$ above and the generating polynomial $S(X) = \sigma_0 + \sigma_1 X + \ldots + \sigma_{p^r-1}X^{p^r-1}$ of σ.

Proposition 2. *Let* σ *be the binary sequence of period* $q = p^r$ *with odd* $r \ge 3$ *defined in Equation* (1) *and let* $S(X)$ *be the generating polynomial of* σ. *Then for* $1 \le m \le r$ *we have*

$$S(X) \equiv \mathcal{A}^{(m)}(X) \pmod{X^{p^m} - 1}.$$

Proof. Write

$$S(X) \equiv \sum_{i=0}^{p^m-1} b_i X^i \pmod{X^{p^m} - 1}.$$

Then we have

$$b_i \equiv \sigma_i + \sigma_{i+p^m} + \cdots + \sigma_{i+(p^{r-m}-1)p^m} \pmod 2.$$

Let $i = i_1 + i_2 p + \ldots + i_m p^{m-1} < p^m$ with $0 \le i_1, i_2, \ldots, i_m < p$ and $\rho_i = i_1\gamma_1 + i_2\gamma_2 + \ldots + i_m\gamma_m$. For $t = t_{m+1} + t_{m+2}p + \ldots + t_r p^{r-m-1} < p^{r-m}$ with $0 \le t_{m+1}, t_{m+2}, \ldots, t_r < p$, we compute

$$(-1)^{b_i} = (-1)^{\sum_{t=0}^{p^{r-m}-1} \sigma_{i+tp^m}} = \sum_{t_{m+1}=0}^{p-1} \cdots \sum_{t_r=0}^{p-1} \chi(\rho_i + t_{m+1}\gamma_{m+1} + \ldots + t_r\gamma_r) = \phi^{(m)}(\rho_i) = (-1)^{a_i^{(m)}}.$$

So we complete the proof. □

Theorem 1. *Let* σ *be the binary sequence of period* $q = p^r$ *with odd* $r \ge 3$ *defined in Equation* (1). *If 2 is a primitive root modulo* p^2, *the linear complexity of* σ *over* $\mathbb{F}_2$ *satisfies*

$$LC^{\mathbb{F}_2}(\sigma) = \begin{cases} p^r, & \text{if } p \equiv 3 (\text{mod } 4), \\ p^r - 1, & \text{if } p \equiv 1 (\text{mod } 4). \end{cases}$$

Proof. According to Equation (2) and

$$X^{p^r} - 1 = (X-1) \prod_{j=0}^{r-1} \left(1 + X^{p^j} + X^{2p^j} + \ldots + X^{(p-1)p^j}\right),$$

we only need to consider

$$\gcd(X-1, S(X)) \text{ and } \gcd(\Phi_j(X), S(X)),$$

where $\Phi_j(X) = 1 + X^{p^j} + X^{2p^j} + \ldots + X^{(p-1)p^j}$ for $0 \leq j < r$. $\Phi_j(X)$ is irreducible since 2 is a primitive root modulo p^2.

We suppose $\Phi_{j_0}(X)|S(X)$ for some $0 \leq j_0 < r$, then $\Phi_{j_0}(X)|\mathcal{A}^{(j_0+1)}(X)$ by Proposition 2. If we write for some $H(X)$

$$\mathcal{A}^{(j_0+1)}(X) = \Phi_{j_0}(X)H(X),$$

then $wt(\Phi_{j_0}(X)H(X))$ is divided by p, but $wt(\mathcal{A}^{(m)}(X)) = (p^m-1)/2 + \delta_m$ by Proposition 1, a contradiction. So we derive $\gcd(\Phi_j(X), S(X)) = 1$ for all $0 \leq j < r$. Finally, since $S(1) = (p^r-1)/2$ it is easy to get $\gcd(X-1, S(X))$ depending on $p \pmod 4$. □

We remark that, Theorem 1 is not true if 2 is not a primitive root modulo p^2. For example, let $q = 7^3$ (*i.e.*, $p = 7$ and $r = 3$) and $\mathbb{F}_q = \mathbb{F}_p(\alpha)$, where α is a root of primitive polynomial $X^3 + 3X + 2$. For the basis $1, \alpha, \alpha^2$, we have $S(X) = X^4 + X^2 + X \pmod{X^7 - 1}$. Thus $X^3 + X + 1$ divides $\gcd(S(X), X^q - 1)$. Indeed in this case, $LC^{\mathbb{F}_2}(\sigma) = 340 (< q = 343)$.

4 k-Error Linear Complexity

In this section, we prove the k-error linear complexity of σ over $\mathbb{F}_2$ for small k.

Theorem 2. *Let σ be the binary sequence of period $q = p^r$ with odd $r \geq 3$ defined in Equation (1). If 2 is a primitive root modulo p^2, then for $1 \leq k < (p-1)/2$ the k-error linear complexity of σ over $\mathbb{F}_2$ satisfies*

$$LC_k^{\mathbb{F}_2}(\sigma) = p^r - 1.$$

Proof. Let $e(X)$ be an error corrected sequence and $wt(e(X)) < (p-1)/2$. It is clear that $S(X) - S(1)$ is divided by $X - 1$ and $LC_1^{\mathbb{F}_2}(\sigma) \leq p^r - 1$.

Further, if we suppose that $S(X) + e(X)$ is divided by $\Phi_m(X) = X^{(p-1)p^{m-1}} + \cdots + X^{p^{m-1}} + 1$ for some $1 \leq m \leq r$ then as in the proof of Theorem 1 we obtain

$$\mathcal{A}^{(m)}(X) + e(X) \pmod{X^{p^m} - 1} = \Phi_m(X)H(X)$$

for some $H(X)$, then $wt\left(\mathcal{A}^{(m)}(X) + e(X) \pmod{X^{p^m} - 1}\right)$ is an integer divided by p, but $wt(\mathcal{A}^{(m)}(X)) = (p^m - 1)/2 + \delta_m$ by Proposition 1 and $wt(e(X) \pmod{X^{p^m} - 1}) < (p-1)/2$, a contradiction. So, $\gcd(S(X) + e(X), X^q - 1) = X - 1$ and $LC_k^{\mathbb{F}_2}(\sigma) = p^r - 1$ for $1 \leq k < (p-1)/2$. □

For $k \geq (p-1)/2$, it seems not easy to determine the k-error linear complexity. Below, we list some examples when $r = 3$.

Let $\Delta = \min\left(wt\left(U_1^{(2)}(X)\right), wt\left(U_g^{(2)}(X)\right)\right)$. Then $\Delta \leq (p-1)/2$. By Equation (4) and Proposition 2 it is clear that for k from $(p-1)/2$ to $(p^2-1)/2$ the values of the k-error linear complexity of σ depend on Δ. The sum $S(X) + e(X)$ can be divided by $X^{p(p-1)} + \cdots + X^p + 1$ for some $e(X)$ with k terms for $k \geq (p-1)\Delta + (p-1)/2$. However the value of Δ is from 0 to $(p-1)/2$ and depends on the choice of the basis.

Let $\mathbb{F}_{p^3} = \mathbb{F}_p(\alpha)$, where α is a root of primitive polynomial $p(X)$.

1) Suppose $p = 5, p(X) = X^3 + 3X + 2$ or $p = 13, p(x) = x^3 + x^2 + 7$. We check that

$$LC_k^{\mathbb{F}_2}(\sigma) = \begin{cases} p^3 - 1, & \text{if } 0 \leq k < (p-1)/2, \\ p^3 - p, & \text{if} \\ & (p-1)/2 \leq k < p(p-1)/2, \\ p^3 - p(p-1) - 1, & \text{if} \\ & p(p-1)/2 \leq k < (p^2-1)/2. \end{cases}$$

2) Suppose $p = 11, p(x) = x^3 + x + 4$. we have

$$LC_k^{\mathbb{F}_2}(\sigma) = \begin{cases} p^3, & \text{if } 0 \leq k < 5, \\ p^3 - p, & \text{if } 5 \leq k < 45, \\ p^3 - p(p-1) - 1, & \text{if } 45 \leq k < 61. \end{cases}$$

5 Conclusions and Final Remarks

We studied the linear complexity of binary sequences of period p^r derived from the discrete logarithm in finite fields for odd r. We found the exact values of the linear complexity of the sequences for odd $r \geq 3$ and proved some partial results about the k-error linear complexity. To complete the work, we make a few remarks on the even case for $r > 3$.

For even $r > 2$, we also have $S(X) \equiv \mathcal{A}^{(m)}(X) \pmod{X^{p^m} - 1}$ for the generating polynomial of σ. But, in this case $\chi(g) = 1$, which leads to $wt\left(V^{(m)}(X)\right) \in \{0, p-1\}$ and $wt\left(U_j^{(m)}(X)\right) = wt\left(U_{g*j}^{(m)}(X)\right)$ for any j from 1 to $p^{m-1} - 1$. Hence Lemma 3(iii) and Proposition 1 are not true for even $r > 2$. Further, we see here that $S(X) \pmod{X^p - 1} = wt\left(U_1^{(2)}(X)\right) \sum_{j=1}^{p-1} X^j$ and hence $LC^{\mathbb{F}_2}(\sigma) \leq p^r - p + 1$.

Examples indicate that even with small values of k, calculating k-error linear complexity of σ for even r seems to be a difficult task.

Let $\mathbb{F}_{p^4} = \mathbb{F}_p(\alpha)$, where α is a root of primitive polynomial $p(X)$. We choose $1, \alpha, \alpha^2, \alpha^3$ as the basis.

1) Suppose $p = 5$ and $p(X) = X^4 + X^2 + 2X + 2$. Then we have $wt\left(U_1^{(2)}(X)\right) = 2, wt\left(V^{(m)}(X)\right) = 4$ and check that $LC_9^{\mathbb{F}_2}(\sigma) = 5^4 - 20 = 605 (= p^4 - (p-1)p)$.

2) Suppose $p = 5$ and $p(X) = X^4 + X^3 + X + 3$. Then we have $wt\left(U_1^{(2)}(X)\right) = 4, wt\left(V^{(m)}(X)\right) = 0$ and check that $LC_4^{\mathbb{F}_2}(\sigma) = 5^4 - 20 = 605 (= p^4 - (p-1)p)$.

3) Suppose $p = 11$ and $p(X) = X^4 + X + 2$. Then we have $wt\left(U_2^{(2)}(X)\right) = 2, wt\left(V^{(m)}(X)\right) = 10$ and check that $LC_{21}^{\mathbb{F}_2}(\sigma) = 11^4 - 110 = 14621 (= p^4 - (p-1)p)$.

Acknowledgments

Parts of this work were written during a very pleasant visit of Zhixiong Chen to the Novgorod State University in August 2019. He wishes to thank the host for the hospitality.

V. Edemskiy and S. Garbar were supported by RFBR-NSFC according to the research project No. 19-51-53003. Z. Chen was partially supported by the Project of International Cooperation and Exchanges NSFC-RFBR No. 61911530130 and NSFC No 61772292.

References

[1] Z. Chen, V. Edemskiy, P. Ke, and C. Wu, "On k-error linear complexity of pseudorandom binary sequences derived from euler quotients," *Advances in Mathematics of Communications*, vol. 12, no. 4, pp. 805–816, 2018.

[2] Z. Chen and Q. Wang, "On the k-error linear complexity of binary sequences derived from the discrete logarithm in finite fields," *Cryptography and Security*, 2019, (`https://arxiv.org/abs/1901.10086`)

[3] T. W. Cusick, C. Ding, and A. Renvall, *Stream Ciphers and Number Theory*, 1998. ISBN: 9780080541846.

[4] I. Damgård, "On the randomness of legendre and jacobi sequences," in *Conference on the Theory and Application of Cryptography*, vol. 403, pp. 163–172, 1990.

[5] C. Ding, "Pattern distributions of legendre sequences," *IEEE Transactions on Information Theory*, vol. 44, no. 4, pp. 1693–1698, 1998.

[6] C. Ding, G. Xiao, and W. Shan, *The Stability Theory of Stream Ciphers. Lecture Notes in Computer Science*, 1991. ISBN: 978-3-540-54973-4.

[7] V. Edemskiy, C. Li, X. Zeng, and T. Helleseth, "The linear complexity of generalized cyclotomic binary sequences of period p^n," *Designs, Codes and Cryptography*, vol. 87, pp. 1183–1197, 2019.

[8] J. H. Kim and H. Y. Song, "Trace representation of legendre sequences," *Designs, Codes and Cryptography*, vol. 24, pp. 343–348, 2001.

[9] W. Meidl and A. Winterhof, "Lower bounds on the linear complexity of the discrete logarithm in finite fields," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2807–2811, 2001.

[10] W. Meidl and A. Winterhof, "On the autocorrelation of cyclotomic generator," *Lecture Notes in Computer Science*, vol. 2948, pp. 1–11, 2003.

[11] A. Sárközy and A. Winterhof, "Measures of pseudorandomness for binary sequences constructed using finite fields," *Discrete Mathematics*, vol. 309, no. 6, pp. 1327–1333, 2009.

[12] M. Stamp and C. F. Martin, "An algorithm for the k-error linear complexity of binary sequences with period 2^n," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1398–1401, 1993.

[13] A. Winterhof, "A note on the linear complexity profile of the discrete logarithm in finite fields," in *Coding, Cryptography and Combinatorics*, vol. 23, pp. 359-367, 2004.

[14] Z. Xiao, X. Zeng, C. Li, and T. Helleseth, "New generalized cyclotomic binary sequences of period p^2," *Designs, Codes and Cryptography*, vol. 86, no. 7, pp. 1483–1497, 2018.

[15] Z. Ye, P. Ke, and C. Wu, "A further study of the linear complexity of new binary cyclotomic sequence of length p^n," *Applicable Algebra in Engineering Communication and Computing*, vol. 30, no. 3, pp. 217–231, 2019.

Biography

Vladimir Edemskiy finished Leningrad University in Mathematics and he received the Ph.D. degree in Algebra and Number Theory from Leningrad University in 1990. In 2010 he received the D. Sc. degree from Novgorod Sate University. Now he is a professor of Novgorod Sate University. His research interests include pseudorandom sequences, design sequences and cryptography.

Zhixiong Chen was born in 1972. He received the M.S degree in Mathematics from Fujian Normal University in 1999 and Ph.D. degree in Cryptography from Xidian University in 2006, respectively. Now he is a professor of Putian University. He worked as a visiting scholar supervised by Prof. Arne Winterhof in Austrian Academy of Sciences (Linz) in 2013 and by Prof. Andrew Klapper in University of Kentucky (Lexington) during 2014-2015, respectively. His research interests include stream cipher, elliptic curve cryptography and digital signatures.

The Sergey Garbar was born in 1986. He received the M.S degree in Applied Mathematics and Computer Science from Novgorod Sate University. Now he is a lecturer of Novgorod State University. His research interests include stochastic modeling and design sequences.

Reversible Data Hiding with Contrast Enhancement Based on Laplacian Image Sharpening

Chengkai Yang[1], Zhihong Li[1], Wenxia Cai[2], Shaowei Weng[3], Li Liu[1], and Anhong Wang[1]
(Corresponding author: Zhihong Li)
Institute of Electronic Information Engineering, Taiyuan University of Science and Technology[1]
Taiyuan 030024, China
College of Mechatronics, Shijiazhuang University, Shijiazhuang, China[2]
School of Information Engineering, Guangdong University of Technology, Guangzhou, China [3]
(Email: zy_lzh@sohu.com)
(Received Oct. 17, 2019; Revised and Accepted Jan. 15, 2020; First Online Feb. 5, 2020)

Abstract

In 2014, Wu *et al.* proposed a reversible data hiding method with contrast enhancement (RDH-CE) that emphasized that the visual quality of the image was more important than having a high peak signal-to-noise ratio (PSNR). But this method focused only on global enhancements and ignored the details. There were more obvious distortions of the visual image as the embedding level increased, and embedding capacity was relatively low when the embedding level was small. Therefore, in this paper, we proposed a new RDH method with contrast enhancement based on Laplacian sharpening. First, the details of the edges of images and the clarity of images were emphasized by Laplacian sharpening, and the visual distortions of the images were reduced by sharpening scale factor. Then, the embedding capacity was increased by combining the difference expansion and digital inverse transformation to apply the operator to all of the pixels in the image. The experimental results demonstrate the effectiveness of the proposed scheme.

Keywords: Reversible data hiding; Laplacian sharpening; Contrast enhancement

1 Introduction

Data hiding has been used extensively in protecting ownership, fingerprinting, authentication and secret communication [9,17,20]. Data hiding can be classified into two categories, *i.e.*, reversible and irreversible data hiding, with the latter usually causing permanent distortion of the image. However, for sensitive images, such as art, military, and medical images as carriers of stored data, permanent damage to the original images is not allowed during the embedding and extraction of information. For example, a patient's record and diagnosis can be embedded into her or his CT image for confidentiality purposes. When a new diagnosis is to be made based on the original image, the hidden data have to be extracted, and the original image has to be recovered losslessly. This requires reversible data hiding (RDH) technology that can both extract the embedded bits and restore the original cover image without any error [1,2,4,7,8,10,12,13,15].

RDH algorithms are based mainly on three techniques, *i.e.*, difference expansion (DE) [1,2,7,12,15], histogram shifting (HS) [4,8,10,13,14], and prediction-error (PE) [5,11,18,19], the purposes of which are to provide higher capacity and PSNR. Tian [12] was the first to propose the idea of DE for RDH. The algorithm computed the features of consecutive pixel pairs in the image using a decorrelation operator, and, then, the data were embedded into an expanded version of these features, thereby effectively improving the hiding capacity. However, RDH based on DE usually causes visual distortion in the stego-image when the difference is large. Ni *et al.* [10] proposed the RDH algorithm based on HS. After determining the peak and zero of the bins of the histogram, this algorithm moved the bins between the peak and the zero toward the zero points, and vacated the bins near the peak to embed the information. Although Ni *et al.* is algorithm achieved a significant improvement in the quality of images and ensured higher PSNR values, the embedding capacity was relatively low due to the limitation on the number of peak points. Ou *et al.* [11] proposed an RDH approach based on PE, in which the pixels were sorted according to pixel correlation, and secret information was embedded according to the difference relationship between the minimum and the second minimum value, as well as the maximum and the second maximum values. The algorithm achieved better image quality, but the hiding capacity was low due to the limited effective difference.

It should be noted that these RDH algorithms pay more attention to the improvement of PSNR than to the visual quality of the stego-image. In 2014, Wu *et al.* [16] proposed a reversible data hiding algorithm with contrast enhancement (RDH-CE), and it provided a new direction for research related to RDH. The motivation for the proposed algorithm was that they believed that the improvement of the contrast in an image was more important than maintaining a high PSNR. Although the value of the PSNR is not high in some images, the improvement of the contrast in the image still can maintain the good visual quality of the image. Therefore, stego-images with better contrast are less likely to be suspicious to attacker when they don't know the original cover image. In order to embed information and improve contrast, Wu *et al.* [16] designed an algorithm based on traditional histogram equalization. Their algorithm pushes the pixel values to the two ends of the dynamic range by hiding data to enhance the contrast of the image. However, this algorithm focuses on the global contrast of the image, and it ignores the local contrast and the details of texture. The algorithm does not achieve higher embedding capacity for the histogram distributions that have lower peaks and wider dynamic ranges, and obvious visual distortion of the image appears when the embedding level is high.

In view of the loss of the details of the texture and the low capacity caused by the traditional RDH-CE scheme, we considered that sharpening the image could enhance the details of the image effectively, such as the edges and the textures of the image, while simultaneously increasing the number of bits embedded in all of the pixels. Therefore, we propose an RDH-CE algorithm based on Laplacian sharpening, which can effectively improve the edges and clarity of images, while increasing their embedding capacities. First, the Laplacian response of all of the pixels is calculated according to the Laplacian operator, and, then, the response and secret bits are mixed with the original pixels in order to enhance the visual effects of the image and hide the secret information. The experimental results showed that the algorithm has higher embedding capacity and image sharpening.

The rest of the paper is organized as follows. In the next section, we review the RDH algorithm and the Laplacian operator which are related closely to the proposed algorithm. In Section 3, we introduce the embedding and extraction processes used in the proposed algorithm. In Section 4, we discuss our evaluation of the performance of the proposed algorithm, our conclusions are presented in Section 5.

2 Related Works

2.1 Wu *et al.*'s Scheme

The RDH-CE algorithm proposed by Wu *et al.* [16] is considered to be the first RDH algorithm to improve image contrast. The algorithm embeds data by changing the distribution of image histograms, and it achieves global enhancement of image contrast. For a grayscale image I, the embedding of data bits starts by searching for the two largest peaks in the histogram of the image. If we assume that I_R represents the peak with a larger pixel value and I_S represents the peak with a smaller pixel value, the bit B_k is embedded into the original image by modifying the pixels values i using Equation (1):

$$i' = \begin{cases} i-1, & if \quad i < I_S \\ I_S - B_k, & if \quad i = I_S \\ i, & if \quad I_S < i < I_R \\ I_R + B_k, & if \quad i = I_R \\ i+1, & if \quad i > I_R \end{cases} \tag{1}$$

This implies that the pixel values between I_S and I_R are unchanged, and the histogram bins outside the peak and the second peak shift one pixel toward both ends of the dynamic range, and the bins next to the peak and the second peak are used to embed the information. This scheme identifies the peak and the second peak repeatedly, and then it shifts the bins and embeds information to fill the dynamic range of the histogram. In fact, the effect of pushing pixel values toward the two extremes of the dynamic range is similar to histogram equalization for improving the global contrast of the image. Thus, it is expected to enhance the global contrast of the image.

Pre-processing is required in this scheme to avoiding pixel overflow. The pixel value i is increased by L when $i \in (0, L-1)$, and decreased by L when $i \in (256-L, 255)$, where L is the number of loops of the algorithm, which is determined by the size of the embedded data. The positions of the pixels during pre-processing are recorded using a location map, and the position where the pixel is changed is recorded as 1, and the position where the pixel is unchanged is recorded as 0. Then, the location map is encoded and compressed as extra bits embedded in the image. During this process, the peak and the second peak must be embedded into the image to ensure the extraction of secret bits and the restoration of the image. In the extraction and recovery process, only I_S and I_R are required.

Since only I_S and I_R are required and they can be extracted by the LSB of the image, the embedded bit $B_k^{'}$ is extracted by considering the values of$I_S$and $I_S - 1$ and I_R and $I_R + 1$ in the stego-image, such that

$$B_k^{'} = \begin{cases} 1, & if \quad i' = I_S - 1 \, or \, i' = I_R + 1 \\ 0, & if \quad i' = I_S \, or \, i' = I_R \end{cases} \tag{2}$$

The original pixel values, i, are recovered using

$$i = \begin{cases} i'+1, & if \quad i' \leq I_S - 1 \\ I_S, & if \quad i' = I_S \\ I_R, & if \quad i' = I_R \\ i'-1, & if \quad i' \geqslant I_R + 1 \end{cases} \tag{3}$$

This algorithm can improve the image contrast and embed significant payloads into the stego-image. However, the enhancement that is achieved in the contrast is global enhancement.

2.2 Image Sharpening of Laplace

The Laplace operator is one of the commonly used edge detection and image sharpening operators, and it is described by Equation (4):

$$\begin{aligned}\nabla^2 f(i,j) =& f(i+1,j) + f(i-1,j) + f(i,j+1) \\ &+ f(i,j-1) - 4f(i,j),\end{aligned} \tag{4}$$

where $f(i,j)$ is the pixel value of the digital image, and the Laplace operator also can be represented as a template, as shown in Figure 1.

0	1	0
1	-4	1
0	1	0

(a)

1	1	1
1	-8	1
1	1	1

(b)

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

(c)

Figure 1: Laplace template, (a) Laplace algorithm template; (b) Laplace extension template; (c) Laplace other templates

The sharpening of the image enhances the grayscale contrast of the image by differential operation, and it highlights the details of the image, which makes a blurred image clearer. The Laplace operator also is a differential operator, and it can enhance the region where the image grayscale is interrupted. Therefore, the Laplace operator is selected to perform template convolution on the image, and the Laplace response image that is generated is superimposed on the original image to generate a sharpened image. The basic method is represented by Equation (5):

$$g(i,j) = f(i,j) + k \nabla^2 f(i,j), \tag{5}$$

where $g(i,j)$ is the pixel value of the sharpened image. It can highlight the image edge while preserving its background.

3 Proposed Scheme

The main intention of our scheme is to absorb the visual effect of the image brought about by sharpening the image, thereby enhancing the detailed information of the image and achieving the purpose of RDH. Therefore, by combining Laplace sharpening, difference expansion and digital inverse transformation, we propose a reversible data hiding algorithm with contrast enhancement based on Laplace sharpening.

Figure 2 shows the framework of the algorithm, which consists of two phases. In the embedding phase, three steps are performed, *i.e.*, global image enhancement, Laplace sharpening and data embedding, and extra bits embedding. In the extraction phase, the order of the three steps is reversed to restore the original cover image and the embedded data. The details of these phases are presented in the following subsections.

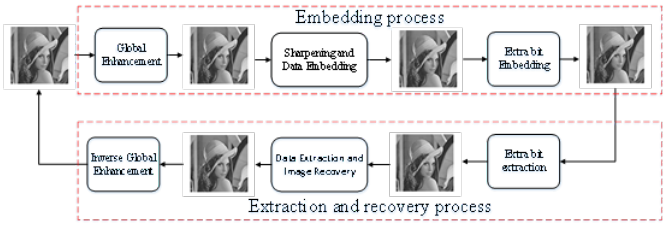


Figure 2: Framework of algorithm

3.1 The Embedding Phase

3.1.1 Pre-Processing of the Image to Achieve Global Enhancement

The embedding process starts with pre-processing the image. For natural pixels of the image that are not distributed over the entire dynamic range, a simple linear stretch is used to achieve global enhancement.

Assuming that H_G, L_G and M_G are the highest, lowest, and average values of the pixel values in an 8-bit $M \times N$ grayscale cover image, I, the pixel value in cover image I are classified first into two groups based on M_G, and the pixel values p in the range $[L_G, M_G - 1]$ are mapped to the range $[0, M_G - 1]$ by using the following transformation function:

$$F_L(p) = \left\lfloor \frac{M_G - 1}{M_G - 1 - L_G} (p - L_G) \right\rfloor \tag{6}$$

Similarly, the pixel values, p, in the range $[M_G, H_G]$ are mapped to the range $[M_G, 255]$ by using the following function:

$$F_H(p) = \lfloor \frac{255 - M_G}{H_G - M_G} (p - M_G) + M_G \rfloor. \tag{7}$$

Because the transformation functions in Equations (6) and (7) strictly are monotonically increasing functions, the two functions are connected seamlessly. Thus, stretching the intensity values using these functions preserves the order of the intensities without any merging between neighboring intensities. However, in order to ensure that the image can be recovered completely after the secret information has been extracted, the parameters of the transformation function must be embedded into the cover image as additional information.

3.1.2 Laplace Sharpening and Data Embedding

The next stage is sharpening and embedding secret data in the globally enhanced image. The following Laplace

0	-1	0
-1	4	-1
0	-1	0

Figure 3: Laplace operator

operator was chosen to ensure higher embedding capacity and reversibility of the algorithm. The processes of Laplace sharpening and embedding data are described as follows:

Input: The pre-processed cover-image, $I^{'}$, sized 512×512, and the randomly generated secret bits, B_k.

Output: The stego-image, LW.

Step 1. As shown in Figure 4, divide the cover-image, $I^{'}$, into region A and region B for embedding the secret bits and unchanging area, respectively, and each small square represents a pixel. For example, given a 512×512 cover image,region A contains 510×510 pixels, and the pixels in the first row and column on the far side of the image, *i.e.*, region B, are not processed.

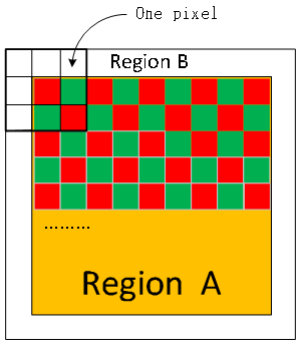


Figure 4: Division of the cover image region

Step 2. Select a 3×3 block from the first pixel of the cover image Region A, as shown in the red area of Figure 4, and perform a Laplace convolution on the block center pixel value (red region), to get the Laplace response of the center pixel $\nabla^2 f(i,j)$.

$$\begin{aligned}\nabla^2 f(i,j) =& 4f(i,j) - f(i+1,j) - f(i-1,j) \\ & - f(i,j+1) - f(i,j-1).\end{aligned} \tag{8}$$

Step 3. Sum the current image pixel and the Laplace response of the point according to Equation (9) to obtain a response pixel, $P(i,j)$.

$$P(i,j) = f(i,j) + \left\lceil \frac{1}{k} \times \nabla^2 f(i,j) \right\rceil, \tag{9}$$

where k is the degree of scaling of the Laplace response. The value of k is determined by the pixel value of the image detail that must be enhanced, as well as the visual effect.

Step 4. Embed the secret bits into the response pixel according to Equation (10) and obtain the watermark sharpening pixel, LW.

$$LW(i,j) = \begin{cases} P(i,j) + 1 + B_k, & if \quad P(i,j) \, mod 2 = 1 \\ P(i,j) + B_k, & if \quad P(i,j) \, mod 2 = 0 \end{cases} \tag{10}$$

where B_k is the sequence of the secret information.

Step 5. Repeat Steps 2 through 4 until all of the blocks have been scanned.

Step 6. Perfect the remaining pixels (green areas) of region A according to the same operation on the sharpened image LW of the above steps to further adjust the image contrast. The selection of the k-value also depends on the scaling required and the visual effect of the image.

3.1.3 Embedding Extra Information

In the aforementioned process, if the result of the pixels exceeds 255 or is less than 0, we must consider the pixel overflow, and the pixel that may overflow is not processed and retains the original value. As shown in Equation (11), the location map (LM) is used to record the position of the pixel. The location map requires code compression for extra information to be embedded into the image.

$$LM(i,j) = \begin{cases} 1, & if \quad LW(i,j) < 0 \; or \; LW(i,j) > 255 \\ 0, & others \end{cases} \tag{11}$$

In this paper, the two parts of extra information, *i.e.*, the parameters of the transformation function and LM, are embedded by pixel value ordering and the prediction error expansion embedding scheme (PVO-k) [11]. This is because the algorithm effectively can embed additional information into the image without affecting the enhanced image. The PVO-k algorithm is not described here, because the embedding of extra bits can be replaced by any RDH algorithm.

3.2 Data Extraction and Image Restoration

The goal of data extraction is to extract the secret bits from the stego-image accurately while ensuring that the cover image is not distorted. In our scheme, the data extraction algorithm is a simple inverse process of data embedding. Its steps are described as follows:

Input: The stego-image LW.

Output: The cover-image, I, and the secret bits, B_k.

Step 1. Extract the extra bit through the inverse process of PVO-k. And obtain the location map and the parameters of the linear transformation function.

Step 2. Divide the stego-image into region A and region B; extraction is started from the green area pixels of region A in Figure 4.

Step 3. The embedded information is extracted according to Equation (12).

$$B_k{}' = LSB\left(LW\left(i,j\right)\right). \tag{12}$$

Step 4. Restore the response pixels using the acquired embedded information, as shown in Equation (13).

$$P' = \left(LW\left(i,j\right) - B_k{}'\right). \tag{13}$$

Step 5. Restore the pixels,$I^{'}(i,j)$,of the point according to Equation (14).

$$\begin{aligned} I'(i,j) = \lfloor \frac{1}{k+4} \times [k \times P' + LW(i+1,j) \\ + LW(i-1,j) + LW(i,j+1) \\ + LW(i,j-1)] \rfloor. \end{aligned} \tag{14}$$

Step 6. The cover image I can be restored completely according to the linear transformation parameters.

3.3 An Example Demonstrating the Process

In this section, we present a simple example of processing the hypothetical image shown in Figures 5 and 6 in order to show the mechanics of the proposed algorithm. In order to guarantee reversibility, the white pixels in the image are not processed.

Start with the first shaded pixel 26, and bring the template into the template using its top, bottom, left, and right values, here the value of k is 2.

Embedding process:

$$\begin{aligned} \nabla^2 f(1,1) &= \lceil \frac{4 \times 26 - 22 - 25 - 24 - 22}{2} \rceil = 6 \\ P(1,1) &= 26 + 6 = 32 \\ B_k &= 1 \\ LW\left(1,1\right) &= 32 + 1 = 33 \end{aligned}$$

Extraction process:

$$\begin{aligned} B_k{}' &= 1 \\ P' &= 33 - 1 = 32 \\ I'\left(i,j\right) &= \left\lfloor \frac{1}{2+4} \times (2 \times 32 + 22 + 25 + 24 + 22) \right\rfloor \\ &= 26. \end{aligned}$$

Red is the hidden pixel of the first layer, and green is the hidden pixel of the second layer. Together they form Region A, and the embedding process is started from the red pixels, and all red pixels are completed before the green pixels. In the extraction, starting from the green pixels, the least significant bits are extracted sequentially, and the original values are restored using Equations (13) and (14); then, the secret bits and the original pixels are extracted and restored according to the restored green pixels until all secret information has been extracted and the original image has been restored.

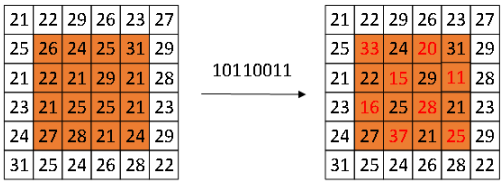


Figure 5: The first layer

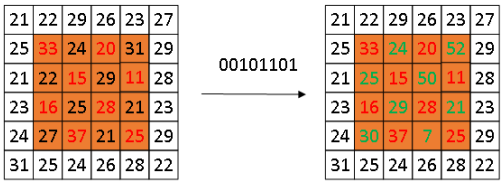


Figure 6: The second layer

4 Experimental Results

The proposed algorithm was compared with the traditional contrast enhancement algorithm proposed by Wu *et al.* [16], the SARDH algorithm proposed by Jafar *et al.* [6] and the PAB algorithm proposed by Chen *et al.* [3]. In the comparison, the SARDH algorithm here also was an RDH algorithm that considered image sharpening. The test images were 512×512 grayscale images of Lena, Baboon, Airplane, and Watch, and they were used to verify the effectiveness of the proposed algorithm.

Figure 7 shows a shadow at the nose of the (b) and (c) images, and the nose of the (d) image is not smooth enough. Figure(e) has better visual effects than the other three figures.

In Figure 8, although the hair in images(b) and (c) is relatively darker in color and the global contrast is strong enough, it is more abrupt in terms of visual effects. Compared with the other three figures, the overall visual effect of image(e) is gentler, and the details of the hair are clearer.

Compared to the other three pictures, the letters on the airplane in Figure 9 are clearer than the letters in image(e). The mountains and clouds in images (b) and (c) are too abrupt, and the details of the mountains and clouds details are clearer in image(d).

Compared to the other three pictures, the picture of the digit on the watch and the buckle in Figure 10 is clearer than that of image(e). As can be seen from the partial detail in Figures 7, 8, 9, and 10, the proposed algorithm can enhance the details at the edges of the image better, and the visual effect of the image is not too abrupt

Figure 7: (a) original, (b) RDH-CE, (c) PAB, (d) SARDH, (e) the proposed scheme

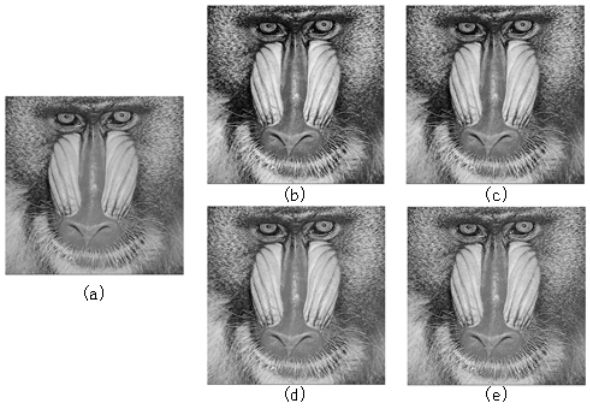


Figure 8: (a) original, (b) RDH-CE, (c) PAB, (d) SARDH, (e) the proposed scheme

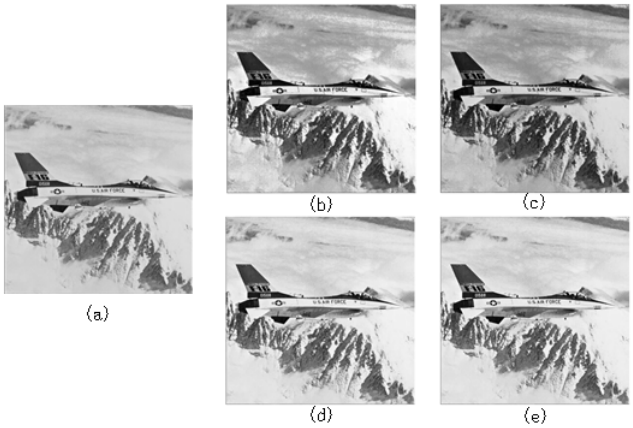


Figure 9: (a) original, (b) RDH-CE, (c) PAB, (d) SARDH, (e) the proposed scheme

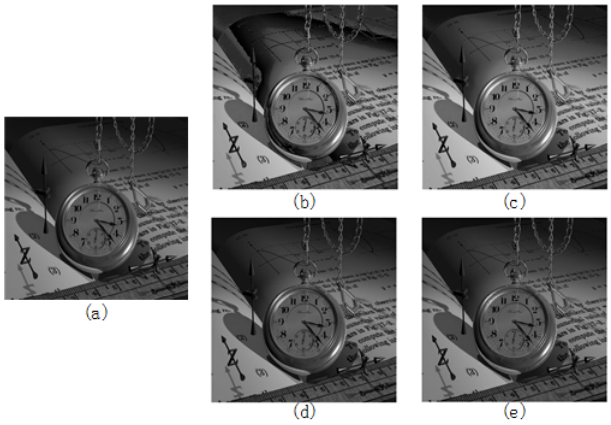


Figure 10: (a) original, (b) RDH-CE, (c) PAB, (d) SARDH, (e) the proposed scheme

after the overall enhancement, making it more consistent with the human visual sense.

Based on the description of the above four experiments, the RDH-CE and PAB schemes mainly improved the global contrast of the images, and the proposed scheme did a better job of enhancing the details at the edges of the images. Thus, the visual effect of the image will not be too abrupt after the overall enhancement, and the image is more consistent with the human visual sense. Thus, the proposed scheme makes up for the shortcoming associated with the enhancement of the details in the images.

In order to evaluate the performance of the proposed scheme further and compare it with the state-of-the-art algorithms, different performance metrics were considered. The gray mean grads (GMG), information entropy (H) and tenegrad measure (TEN) were used to judge the enhancement effect of the images, and the embedding capacity (E) was used to measure the data hiding performance.

1) The gray mean grads (GMG). The GMG value is obtained by squaring the gray value of adjacent pixels in the length and width directions of the image and then determining the root mean square, which can reflect the contrast and texture features of the image. Higher GMG values usually reflect higher clarity and quality in images. The formula used to calculate GMG was:

$$GMG = \frac{1}{(M-1)(N-1)} \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \sqrt{\frac{\Delta I_x^2 + \Delta I_y^2}{2}} \tag{15}$$

Table 1 shows the GMG values of different algorithms. Compared with the other three algorithms, the proposed scheme achieves some enhancement of the details for all pixels of the image, and the gradient value of each pixel is relatively higher. So the proposed scheme has better visual effects concerning the details of the image.

2) Information entropy (H). Information entropy, H,

Table 1: Gray Mean Grads (GMG) of different algorithms

Image	RDH-CE	PAB	SARDH	Proposed
Lena	6.9029	6.4713	8.7163	14.1713
Baboon	24.8951	22.6067	25.9783	57.8684
Airplane	7.7471	7.3489	8.9676	14.2437
Watch	8.7406	8.7406	8.9476	15.9453

represents the average amount of information in an image. Higher H values usually reflect more details in the image. The formula is:

$$H = -\sum_{i=0}^{l-1} p(i)\, log_2 p(i). \tag{16}$$

The data in Table 2 indicate that the H value of the proposed scheme is slightly higher than the H values for other algorithms, which indicates that our proposed algorithm provides more details.

3) The tenegrad measure (TEN). The TEN is a well-known benchmark measure of the sharpness of images, and it is defined by Equation (17). Higher TEN values usually reflect higher contrast levels and stronger edges in images.

$$TEN = \sum_i \sum_j G_{ij}. \tag{17}$$

It is obvious that our algorithm has better performance than the others.

4) The embedding capacity (E). The E is reported as the pure embedding capacity that is normalized by the image size to measure the embedding rate in bits per pixel (bpp) using Equation (18).

$$E = \frac{Embedded\, Bits - Extra\, bit}{M \times N}. \tag{18}$$

The results show that the proposed algorithm significantly improves the number of embedding bits. Compared with the SARDH algorithm, our algorithm provides an embedding capacity closer to 1 bpp, which is far more than the other three algorithms.

5 Conclusion

The RDH-CE scheme, which improves the contrast in images after information has been embedded, gives a new direction for RDH. It can be said that the stego-images that have better contrast are less likely to attract the attention of attackers if they do not know the original image. In this paper, we proposed an RDH-CE algorithm based on image sharpening, which can sharpen the details of images and embed a large amount of secret information. Compared with the traditional, contrast-enhanced RDH algorithm, the proposed algorithm compensates for the details of images that are neglected by the global enhancement. Our experimental results indicated that our proposed algorithm provided better visual effects for the details of images than the other algorithm. Also, our experimental evaluations verified that the proposed algorithm has a large embedding capacity and a relatively better edge enhancement effect.

Acknowledgments

This study was supported by the Shanxi Natural Science Foundation under Grand (No.201801D121129), National Natural Science Foundation of China (No.61672373, No.61501315), Scientifc and Technological Innovation Team of Shanxi Province (No.201705D131025), Key Innovation Team of Shanxi 1331 Project (2017015), Collaborative Innovation Center of Internet+3D Printing in Shanxi Province(201708). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] C. C. Chang, Y. H. Huang, and T. C. Lu, "A difference expansion based reversible information hiding scheme with high stego image visual quality," *Multimedia Tools and Applications*, vol. 76, no. 10, pp. 12659–12681, 2017.

[2] C. C. Chen, Y. H. Tsai, and H. C. Yeh, "Difference-expansion based reversible and visible image watermarking scheme," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8497–8516, 2017.

[3] H. Chen, J. Ni, W. Hong, and T. S. Chen, "Reversible data hiding with contrast enhancement using adaptive histogram shifting and pixel value ordering," *Signal Processing: Image Communication*, vol. 46, pp. 1–16, 2016.

[4] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, "Reversible watermarking based on invariant image classification and dynamic histogram shifting," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 111–120, 2013.

[5] 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*

Table 2: Image information entropy (H) of different algorithms

Image	RDH-CE	PAB	SARDH	Proposed
Lena	7.6231	7.589	7.6459	7.7157
Baboon	7.5601	7.5077	7.6941	7.8814
Airplane	7.1881	7.081	6.9508	7.2376
Watch	7.2642	7.341	7.164	7.3724

Table 3: Tenegrad measure (TEN) of different algorithms ($\times 10^7$)

Image	RDH-CE	PAB	SARDH	Proposed
Lena	1.5086	1.3944	1.6493	1.8695
Baboon	3.8919	3.506	3.9346	5.213
Airplane	1.7147	1.6234	1.8091	2.0154
Watch	1.6651	1.5682	1.7645	1.9008

Table 4: Actual embedding capacity (E) of different algorithms)

Image	RDH-CE	PAB	SARDH	Proposed
Lena	0.2108	0.2217	0.5803	0.9822
Baboon	0.2078	0.1715	0.2863	0.9038
Airplane	0.4795	0.4439	0.6218	0.9788
Watch	0.258	0.2658	0.6847	0.9537

Visual Communication and Image Representation, vol. 49, pp. 351–360, 2017.

[6] I. F. Jafar, K. A. Darabkh, and R. R. Saifan, "Sardh: A novel sharpening-aware reversible data hiding algorithm," *Journal of Visual Communication and Image Representation*, vol. 39, pp. 239–252, 2016.

[7] S. Lakshmanan and M. Rani, "Reversible data hiding in medical images using edge detection and difference expansion technique," *Journal of Computational and Theoretical Nanoscience*, vol. 15, no. 6-7, pp. 2400–2404, 2018.

[8] L. Liu, C. C. Chang, and A. Wang, "Reversible data hiding scheme based on histogram shifting of n-bits planes," *Multimedia Tools and Applications*, vol. 75, no. 18, pp. 11311–11326, 2016.

[9] L. Liu, C. C. Chang, and A. Wang, "Data hiding based on extended turtle shell matrix construction method," *Multimedia Tools and Applications*, vol. 76, no. 10, pp. 12233–12250, 2017.

[10] 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.

[11] B. Ou, X. Li, Y. Zhao, and R. Ni, "Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion," *Signal Processing: Image Communication*, vol. 29, no. 7, pp. 760–772, 2014.

[12] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[13] 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.

[14] 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.

[15] W. Wang, Y. Ye, T. Wang, and W. Wang, "Reversible data hiding scheme based on significant-bit-difference expansion," *IET Image Processing*, vol. 11, no. 11, pp. 1002–1014, 2017.

[16] H. T. Wu, J. L. Dugelay, and Y. Q. Shi, "Reversible image data hiding with contrast enhancement," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 81–85, 2014.

[17] B. Xia, A. Wang, C. C. Chang, and L. Liu, "Reversible data hiding for VQ indices using hierarchical state codebook mapping," *Multimedia Tools and Applications*, vol. 77, no. 16, p. 20519–20533, 2018.

[18] S. Yi, Y. Zhou, and Z. Hua, "Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion," *Signal Processing: Image Communication*, vol. 64, pp. 78–88, 2018.

[19] X. Yu, X. Wang, and Q. Pei, "Reversible watermarking based on multi-dimensional prediction-error ex-

pansion," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 18085–18104, 2018.

[20] Y. Wang, J. Shen, and M. Hwang, "A survey of reversible data hiding for VQ-compressed images," *International Journal of Network Security*, vol. 20, no. 1, pp. 1–8, 2018.

Biography

Chengkai Yang was born in Yuncheng City, Shanxi Province on June 19, 1995. In July 2016, he graduated from the Department of Electronic Information Engineering, Taiyuan University of Science and Technology with a Bachelor of Engineering degree. Now, he is a graduate student in Taiyuan University of Science and Technology. His research direction is reversible data hiding.

Zhihong Li was born in Shanxi Province, P. R. China in 1970. He received the M.E. degree in the Electronic Information Engineering, Taiyuan University of Science and Technology in 1997. His research interest includes image/video coding and secret image sharing.

Wenxia Cai received her B.S. degree and her M.S. degree from North China Electric Power University (Baoding), China respectively in 2003 and in 2006. She is currently a lecturer in Shijiazhuang University. Her research interests include information processing technology and mechatronics, etc.

Shaowei Weng received her Ph.D. degree from the Institute of Information Science at Beijing Jiaotong University in July 2009. She is currently an associate professor in the School of Information Engineering at Guangdong University of Technology. Her research interests include image processing, data hiding and digital watermarking, pattern recognition, computer vision, etc. Now she is in charge of two NSFC (Natural Science Foundation of China) project. In addition, she participates in 973 and 863 projects as the backbone. She publishes more than 20 papers, and applies two national patents.

Li Liu received her B.E. degree in communication engineering in 2002, from Lanzhou Railway University and M.E. degree in communication and information system in 2006, from Lanzhou Jiaotong University. Now, she is a Ph. D student in Northwestern Polytechnical University. Her current research interests include information hiding and secret sharing.

Anhong Wang was born in Shanxi Province, P. R. China in 1972. She received the Ph. D. degree in the Institute of Information Science, Beijing Jiaotong University in 2009. She is now the director of Institute of Digital Media and Communication, Taiyuan University of Science and Technology. Her research interest includes image/video coding and secret image sharing.

Fine-grained Identification for SSL/TLS Packets

Lingjing Kong, Ying Zhou, Guowei Huang, and Huijing Wang
(Corresponding author: Lingjing Kong)

School of Computer Science, Shenzhen Institute of Information Technology
No. 2188, Longxiang Rd, Longgang District, Shenzhen, Guangdong, China
(Email: lingjk11@gmail.com)

(Received June 9, 2019; Revised and Accepted Dec. 3, 2019; First Online Feb. 1, 2020)

Abstract

SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocols have been widely used in data transmission to protect the security and integrity of the data. However, due to the encryption of SSL/TLS, the application data over transmitted packets are invisible and difficult to be distinguished by traditional port-based and DPI(Deep Packet Inspection) ways. Though the method based on statistical features can overcome shortages of the above two ways, it is still hard to achieve fine-grained identification. In this paper, we proposed a solution to extract fingerprint information and then identify the types of flows during handshake phase to avoid inspecting the encrypted data and privacy violation. Besides, two hash tables are built to help fast identify the packets with the same APP ID in the same or among different conversations. Finally, 300 flows are captured and the experiment results show the method is accurate and efficient.

Keywords: Fine-Grained; Identification; SSL/TLS

1 Introduction

For the security and privacy of data transmission, SSL [7]/TLS [6] (Secure Sockets Layer/Transport Layer Security) has been widely applied in the encryption transmission in many aspects(such as e-banking, email, VPN), especially in web security. Because of the encryption by using SSL/TLS, the data transmitted over the Internet are invisible, thus is difficult for network traffic classification and identification. Traffic identification is the significant part for network management, quality of service and network security. The identification of traffic can distinguish the application types of the packets so as to better manage the network, allow or deny bad packets. However, existed methods such as port-based method only identify SSL/TLS packets, but is difficult to identify the concrete types of SSL/TLS packets(For example, Google email type or some companies' VPN types). Even though most enterprises utilize traditional DPI(Deep Packet Inspection) method to achieve identification of SSL/TLS packets, it is not easy to identify non-transparency payload. Besides, brute decryption needs more cost and may also violate the privacy protection.

In view of the above facters, in this paper, we proposed a fine-grained method to identify the SSL application types accurately and fast without touching encrypting information. This approach will be an essential basis for network audit, network management and network security, even for packets label. The main contributions are as follows:

1) To distinguish SSL/TLS packets accurately, we proposed a fine-grained method through fingerprint extraction and matching during the handshake phase.

2) Two hash tables are built so as to fast distinguish packets followed the identified packets in the same flow or among different flows.

The rest of the paper is organized as follows. Section 2 reviews SSL/TLS principle and the related work. Section 3 shows four modules in this methodology and introduces the fine-grained identification algorithm. Section 4 shows the experimental results and discusses the performance of this methodology. Section 5 concludes the whole paper.

2 Related Work

Port-based method is a traditional way to identify the application of packets through recognizing port numbers. These port numbers are usually well-known port numbers registered in IANA (Internet Assigned Numbers Authority) [5] and can be identified by comparing with the records stored in IANA. For example, normally HTTP packets are transmitted by port number 80 and ftp uses port number 21 to transmit data. Port-based method is simple and easy to realize, but not reliable. More flows may not use well-known port numbers to perform data transmission, and some kind of flows may use dynamic port numbers to establish conversations such as P2P flows [1, 3, 11]. Even, to avoid the firewalls certain flows hide their port numbers. All these above make port-based method unreliable.

DPI(Deep Packet Inspection) is a method by inspecting the content of packets payloads and find out the fingerprint information to determine the application types of flows. It is widely applied in companies and has been proved accurate and reliable [4, 12]. However, traditional DPI methods only perform coarse identification for TLS/SSL [3]. Because the payloads of the packets are not transparent and clear after encryption. If using brute force techniques, it will be costly and may sometimes violate the private laws. So traditional DPI is difficult to realize fine-grained identification for TLS/SSL.

Different from DPI, the method based on statistical characteristics of Internet traffic do not inspect the content of packets and mainly foucus on extracting the statistical features of packets or flows [2, 8, 10, 13, 14]. Commonly, these features describe the characteristics of packets behaviors or flow behaviors. They can be packets size, packets intervals or durations of flows, *et al.* It works when facing TLS/SSL packets because it can utilize the statistics to roughly distinguish SSL/TLS packets, but fine-grained identification is also a tough task.

In this paper, we propose a fine-grained identification methodology, which can identify the types of SSL packets accurately and avoid the privacy problem. And through the establishment of two hash tables, the speed of identification is also improved.

3 SSL/TLS Background

3.1 SSL/TLS Overview

SSL is a protocol above TCP layer that utilizes encryption technology to guarantee the security of transmitted data and avoid hijack by the third party. TLS is the subsequent protocol following SSL 3.0 (the latest version of SSL). They are all used to protect the security and integrity of the data. SSL/TLS layer is based on TCP/IP structure, which can be clearly seen in Figure 1.

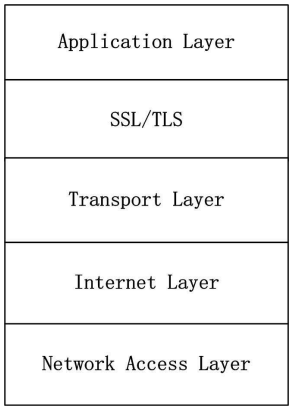


Figure 1: SSL/TLS layer

From Figure 1 we can learn that, SSL/TLS works over the transport layer, and the protocol information can be analyzed and extracted above this layer. SSL/TLS includes two phases: Handshake phase and application data transmission phase. Handshake phase starts before application data transmission phase to validate the identity of two communication endpoints, negotiate encryption algorithms and exchange keys. The data transmitted in handshake phase is transparent and easily identified and extracted.

In this paper, the identification module is performed in this phase and the process of handshake phase will be showed in the next part.

3.2 Handshake

The process of handshake phase can be seen in Figure 2.

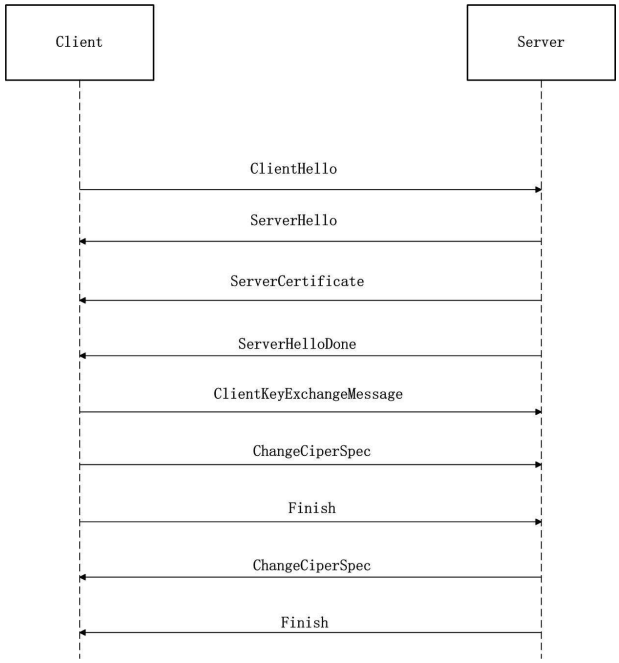


Figure 2: Process of handshake phase

In the handshake phase, there are commonly the following steps to finish handshake session which can be seen in Figure 2:

Step 1. The client firstly sends ClientHello message to start handshake session.

Step 2. The server sends to ServerHello to respond the client. Then the server will send X.509 certificate [9] to the client. And a ServerKeyExchange message and CertificateRequest message may be sent in some cases. Then the client will send ServerHelloDone message to finish the hello phase.

Step 3. The client sends the ClientKeyExchange message to the server and immediately follows the ChangeCiperSpec. Then the client sends finish messeage to complete the handshake session.

Step 4. The server sends ChangeCiperSpec message to the client.At this time,the handshake session is finished. After that the application data begin to be transmitted.

From Figure 2, we noticed that the second SSL/TLS packet from server to client, the server sends the ServerCertificate message to the client. This message includes X.509 certificate where the fingerprint information can be extracted, transformed as features and used to identify the types of TLS/SSL packets. The fingerprint information can be the organization of the application publisher, the department of the application publisher and the purpose of application and so on.

4 The Proposed Method

4.1 The Model of The Proposed Method

Different from most other network flows, SSL flows begin to transmit encrypted data after communication tunnel established in handshake phase. For SSL flows, the packet payloads are transparent in this stage which belongs to the early stage in the whole communication. In this paper, the identification is performed in this early stage and utilizing transparent information in ServerCertificate message and cached hash tables to achieve fine-grained and fast identification.

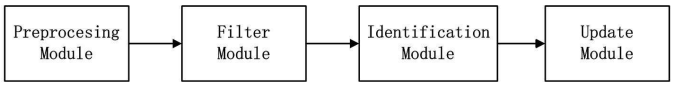


Figure 3: Four modules of fine-graind identification method

The proposed model is mainly composed of four modules: Preprocessing module, filter module, identification module and update module as seen in Figure 3.

Preprocessing module: Two cached hash tables are built to store hash values of APP ID for flows having been identified. Thus it can be used to fast identify packets with the same APP ID through matching with the items stored in these two tables.

Filter module: Create predefined rules to identify if a flow is TLS/SSL flow or not.

Identification module: Extract fingerprint information from the SeverCertificate message and transformed identification features. A newly flows can be identified through computing the similarity with the features.

Update module: Compute the APP ID of newly identified flows and update to the two hash tables.

4.2 Preprocessing Module

In this module, two hash tables will be built. The first hash table H_1 stores the mapping relationships between application types and corresponding application ID (Identification) numbers , that is $<APPID_1, Type>$. *Type* is application types and $APPID_1$ is the corresponding ID numbers within the same conversation(network flow) which can be computed by five-tuple elements(source IP address, destination Ip address, source port, destination port and transport protocol). All items recorded in H_1 infact indicate the packets in the same flow with the same application types.

H_2 indicates the application of packets from different flows and stores mapping relationships $<APPID_2, Type>$. $APPID_2$ can be computed by four-tuple elements (source IP address, destination Ip address, destination ports and transport protocol) because different flows with the same application types have the same four tuples. The items recorded in H_2 indicate the packets in different flows with the same application types.

H_1 and H_2 are utilized to process oncoming packets. When a packet comes, get $APPID_1$ of the packet and query H_1 to check if there is a matching item in H_1. If the matching result is true, the application type can be directly obtained; If the result is false, query H_2. If there is a matching item in H_2, then the application type can be obtained; if not, perform futher identification in the filter module and identification module.

This preprocessing module can fast the packets distinguishing which have been identified and avoid unnecessary work for them.

4.3 Filter Module

Filter module aims at identifying whether a packet is a SSL packet based on the predefined rule. The predefined rule describes the features of the packets and can be expressed as follows:

$$\begin{aligned} filterRule \quad = \quad & \{dir = 0, count = 1, dstport = 443\} \\ & \&\&\{dir = 0, count = 3, offset = 0, \\ & feature = 0 \times 16\}. \end{aligned}$$

In the formula *dir* represents the transmission direction of the packet. If $dir = 0$, it indicates the data is transmitted from the client to the server. If $dir = 1$, it indicates the data is transmitted from the server to the client. *count* indicates the number of the packet located in the whole flow compared with all the other packets in the same flow. *dstport* is the destination port number and offset is the offset in the payload of this packet, while $feature$ is the fingerprint information of the offset.

So $dir = 0, count = 1, dstport = 443$ refers to the port number of the first packet from the client to the server is 443. $dir = 0, count = 3, offset = 0, feature =$ 0x16 refers to the payload offset of the third packet from the client to the server is 0, and starting from the first byte in the third packet, the fingerprint information is 0x16.

4.4 Identification Module

In this module, the fingerprint information will be extracted from the payload of predetermined packets (PrePks), and compare with the records in the feature library. According to the comparsion result, the specific types of PrePks can be get, and the mapping relationships $<APPID_1,\ Type>$ and $<APPID_2,\ Type>$ will be updated in H_1 and H_2. Then the following packets after PrePks will be directly identified through filter module.

Predetermined packets refers to the first five packet($dir = 1, count = 5$) from the server to the client for exchanging X.509 certificates in the handshake phase. Fingerprint information is the features that can be used to identify the specific application types of packets including $countryName$, $stateOrProvinceName$, $localityName$, $organizationName$, $organizationalUnitName$ and $commonName$.

The fingerprint information and the related location in the packets' payload are stored in the linked list as binary numbers. Then the similarity between linked list and the items recorded in feature library is computed:

$$Sim(F, F_k) = \sum_{j=1}^{n} |f_j - f_{kj}|.$$

In the above formula, F is the linked list(infact a vector including n elements), F_k is the kth record in the feature library(also a vector including n elements), f_j the jth bit of F, f_{kj} is the jth bit of the kth record.

If $Sim(F, F_k) = 0$, the application type of predetermined packets is the corresponding type of the kth feature record.

4.5 Update Module

Based on the identification module, new application types are identified and the corresponding $APPID_1$ and $APPID_2$ are computed, and then new mapping relationships will be come up. Finally, H_1 and H_2 are updated by new mapping relationships. The subsequent packets with the same $APPID_1$ and $APPID_2$ can be fast identified by the latest H_1 or H_2.

4.6 Fine-grained Identification Algorithm

TLS/SSL packets get fast and accurate identification through preprocessing module, filter module and identification module. Here, the concrete identification algorithm will be given in Algorithme 1.

The identification procedure can also be described in Figure 4.

Algorithm 1 fine-grained identification algorithm

```
Require: The packet Pk
Begin
Compute APPID_1 of Pk, get (APPID_1)_pk
for i = 1 to |H_1|
if (APPID_1)_pk == (APPID_1)_i
The application type of Pk is (Type)_i
else Compute APPID_2 of Pk, get (APPID_2)_pk
for j = 1 to |H_2|
if (APPID_2)_pk == (APPID_2)_j
The application type of Pk is (Type)_j
else Perform filterRule
Extract the fingerprint information
Compute the Similarity
Determine the application type
Update the mapping relations to H_1and H_2
End
```

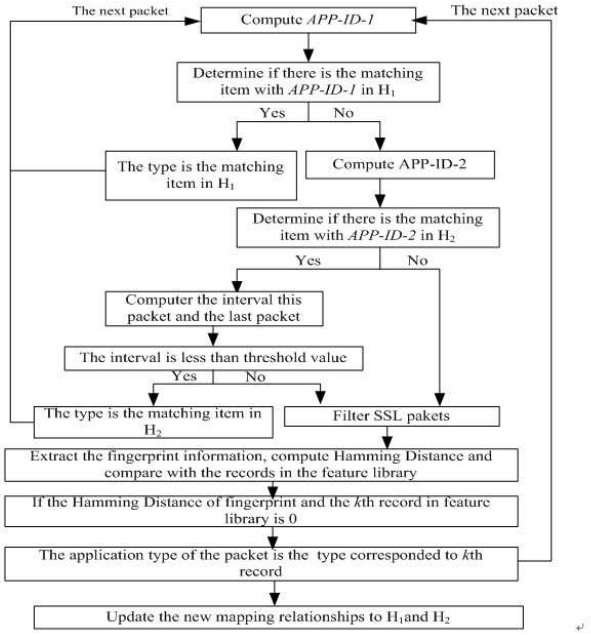


Figure 4: fine-grained identification algorithm

5 Experiment and Analysis

5.1 Dataset

In this paper, to validate this methodology, we captured 300 flows including 10 types TLS/SSL flows including e-mail, e-banking, e-commence, VPN, *et al*, which can be seen in Table 1.

For each type, 30 flows are captured and totally 300 flows are collected. In the whole dataset, 60% of flows are used for model training, while 40% are used for model testing.

Table 1: TLS/SSL traffic types

Application	organizationName	commonName
QQ	Shenzhen Tecent System company	antibot.qq.com
WeChat	Shenzhen Tecent Computer Systems Compan	*.wx.qq.com
VPN	\346\267\261\345\234\263\344\277\241\346	*.sziit.edu.cn
QQ Email	Shenzhen Tecent System company	*.mail.qq.com
Google Browser	Google LLC	*.googleapis.com
Taobao	Alibaba(China) Technology Co., Ltd.	*.taobao.com
Industrial and Commercial Bank of China	Software Development Center	mybank.icbc.com.cn
163 Email	NetEase(Hangzhou)Network Co., Ltd	*.163.com
Xiami Music	Alibaba(China) Technology Co.,Ltd.	*.xiami.net
Youku	Alibaba(China) Technology Co., Ltd.	*.youku.com

5.2 Fingerprint Extraction and Feature Representation

The fingerprint information is extracted from X.509 certificate, and includes: $countryName$, $stateOrProvinceName$, $localityName$, $organizationName$, $organizationalUnitName$ and $commonName$.

All these fingerprint information are transformed to Hexadecimal numbers as features to identify the application types. For example, if the application is 163 mail, the fingerprint information are:

$countryName = CN$
$stateOrProvinceName = Zhejiang$
$localityName = Hangzhou$
$organizationName = NetEase(Hangzhou)$
$NetworkCo., Ltd$
$organizationalUnitName = MAILDept.$
$commonName = *.mail.163.com.$

The corresponding hexadecimal numbers of fingerprint information can be seen in Figure 5.

```
31 0b 30 09 06 03 55 04  06 13 02 43 4e 31 11 30
0f 06 03 55 04 08 13 08  5a 68 65 6a 69 61 6e 67
31 11 30 0f 06 03 55 04  07 13 08 48 61 6e 67 7a
68 6f 75 31 2c 30 2a 06  03 55 04 0a 13 23 4e 65
74 45 61 73 65 20 28 48  61 6e 67 7a 68 6f 75 29
20 4e 65 74 77 6f 72 6b  20 43 6f 2e 2c 20 4c 74
64 31 13 30 11 06 03 55  04 0b 13 0a 4d 41 49 4c
20 44 65 70 74 2e 31 17  30 15 06 03 55 04 03 0c
0e 2a 2e 6d 61 69 6c 2e  31 36 33 2e 63 6f 6d
```

Figure 5: Hexadecimal numbers

Then through matching to the feature library to judge if the application packets belongs to 163 mail application.

5.3 Evaluation

For evaluating the method, four items are involved: True Positive (TP),True Negative(TN),False Positive(FP) and False Negative(FN). They are usually used to evaluate the results of traffic identification.

- True Positive: The type of the flow is X, and the prediction result is also X.
- True Negative: The type of the flow is not X, and the prediction result is not X.
- False Positive: The type of the flow is not X, and the prediction result is X.
- False Negative: The type of the flow is X, and the prediction result is not X.

The accuracy is the most useful indicator values, which can evaluate the good or bad of the identification method. It can be calculated as follows:

$$Accuracy = (TP + TN)/(TP + FP + TN + FN).$$

Through fine-grained identification algorithm, we got 98.4% result which show good performance. It has been proved this method is very effective to identify the specific types of SSL/TLS packets.

After the identification module, the APP ID will be recorded in the two hash tables. hash tables are recorded as $< index, value >$. If the application is 163 mail, $h(x)$ is hash function, $binary(x)$ is the binary transform function, in hashtable h_1,

$x = \{192.168.21.160, 163.177.151, 110, 58473, 443, 6\}$
$index = binary(x)$
$value = 5$
In hashtable h_2,
$x = 192.168.21.160, 163.177.151, 110, 443, 6$
$index = binary(x)$
$value = 5$

"5" represents the value of 163 mail type. Other packets with the same APP ID can be fast identified through hash computing.

5.4 Compasion with Traditional Methods

Compared with traditional methods, our method realized a more concrete and fine-grained distinguishment for TLS/SSL packets, which help better control and manage network. Besides, because of the fast running of hash

function and search, it only takes tens of milliseconds for hash computation. The building of two hash tables reduces redundant work and speed up the identification work. All these help improve the accuracy and increase the efficiency.

6 Conclusions

In this paper, for TLS/SSL network flows identification, the fingerprint information in handshake phase is transparent, and easy extracted to distinguish different types of TLS/SSL flows. It also avoids violating privacy laws. Besides, two hash tables are built and network flows with the same APP ID can be fast identified by comparing with the items in hash tables. Finally, 300 flows for 10 TLS/SSL types are captured and the experiment shows the proposed method can achieve fine-grained and fast identification for TLS/SSL traffic.

Acknowledgments

This study was supported by Natural Science Foundation of Guangdong Province (2018A030310664, 2018A0303130055), Shenzhen Educational Science Planning Project (Grant No. ybfz18279), Fundamental Research Project in the Science and Technology Plan of Shenzhen (Grant No. JCYJ20160527101106061).

References

[1] N. Basher, A. Mahanti, and A. Mahanti,*et al.*, "A comparative analysis of web and peer-to-peer traffic," in *Proceedings of the 17th International Conference on World Wide Web*, pp. 287–296, Apr. 2008.

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," in *IEEE Communications Surveys & Tutorials*, pp. 1153–1176, 2016.

[3] Z. G. Cao, G. Xiong, and Y. Zhao,*et al.*, "A survey on encrypted traffic classification," in *International Conference on Applications and Techniques in Information Security*, pp. 73-81, 2014.

[4] T. Choi, C. Kim, and S. Yoon, *et al.*, "Content-aware internet application traffic measurement and analysis," in *Network Operations and Management Symposium*, pp. 511–524, Apr. 2004.

[5] M. Cotton, L. Eggert, and J. Touch,*et al.*, *Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry*, RFC 6335, 2011.

[6] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246, 2008.

[7] A. Freier, P. Karlton, and P. Kocher, *The Secure Sockets Layer Protocol Vers. 3.0*, RFC6101, 2011.

[8] M. A. Guvensan B. Yamansavascilar and A. G. Yavuz,*et al.*, "Application identification via network traffic classification," in *International Conference on Computing, Networking and Communications*, pp. 843–848, 2017.

[9] R. Housley, W. Polk, and W. Ford,*et al.*, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 3280, Apr. 2002.

[10] A. H. Laskari I. Sharafaldin and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *International Conference on Information Systems Security and Privacy*, pp. 108–116, 2018.

[11] A. Madhukar and C. L. Williamson, "A longitudinal study of P2P traffic classification," in *Modeling, Analysis, and Simulation on Computer and Telecommunication Systems*, pp. 179–188, Oct. 2006.

[12] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," *Passive and Active Network Measurement*, pp. 41–54, 2005.

[13] A. Proto, L. A. Alexandre, and M. L. Batista,*et al.*, "Statistical model applied to netflow for network intrusion detection," in *Transactions on Computational Science XI*, pp. 179–191, 2010.

[14] X. Yu M. Shafiq and A. A. Laghari,*et al.*, "Network traffic classification techniques and comparative analysis using machine learning algorithms," in *IEEE International Conference on Computer and Communications*, pp. 2451–2455, 2016.

Biography

Lingjing Kong received PhD degree from Southwest Jiaotong University, Sichuan, China, 2015. During 2013 to 2014, she also joined in University of Adelaide as a joint PhD student. She is currently a lecturer in Shenzhen Institute of Information Technology, Shenzhen, China. Her research area is network data analysis and machine learning.

Ying Zhou received PhD degree from Sun Yat-sen University, Guangzhou, China, 2014. In 2014, she joined Shenzhen Institute of Information Technology,Shenzhen, China. Her current research interests include local search algorithms and their applications, multiobjective optimization and other evolutionary computation techniques.

Guowei Huang received PhD degree from Nankai University, Tianjin, China, 2009. He is currently a associate professor in Shenzhen Institute of Information Technology, Shenzhen, China. His research area is distributed computer system and streaming media.

Huijing Wang received PhD degree from University of Science and Technology of China, An Hui, China, 2006. She is currently a associate professor in Shenzhen Institute of Information Technology, Shenzhen, China. Her research area is image fusion and enhancement.

A Blockchain-based Privacy-Preserving Authentication Scheme with Anonymous Identity in Vehicular Networks

Liang Wang, Dong Zheng, Rui Guo, ChenCheng Hu, and ChunMing Jing
(Corresponding author: Liang Wang)

National Engineering Laboratory for Wireless Security
School of Cyberspace Security, Xi'an University of Posts and Telecommunications
Xi'an 710121, China
(Email: wangliang_zjk@163.com)

(Received May 5, 2019; Revised and Accepted Dec. 12, 2019; First Online Feb. 1, 2020)

Abstract

With the rapid development of mobile network technology, Vehicular ad-hoc Networks (VANETs), one of the most promising applications in the smart transportation systems, have drawn widespread attention. Unfortunately, authentication and privacy protection of users have seriously restricted the development of VANETs. The past works used to allow a centralized trusted authority to distribute identity information and maintain the operation of the whole system lacking of distributed and decentralized security. In this paper, we propose an authentication scheme based on consortium blockchain with anonymous identity in VANETs. First, when authenticating and providing services, our scheme allows the vehicles using Pseudo IDs obtained from the Road Side Unit (RSU) to protect the privacy of the vehicles preventing location tracking due to disclosure of information. Second, based on consortium blockchain technology, it provides a decentralized, secure and reliable database for storing certificates and the pointer to storage location, which is maintained by the multiple Trusted Authorities (TAs) and RSUs. Furthermore, in the revocation, the RSUs are able to determine promptly that the vehicle has been revoked by adding a revocation tag to the pseudo ID instead of searching the entire certificate revocation list (CRL). According to the security and performance analysis, our scheme owns higher security and efficiency.

Keywords: Anonymity; Blockchain; Privacy-preserving; Revocation; Vehicular Ad-Hoc Networks (VANETs)

1 Introduction

Recently, with the rapid development of the automobile industry and Internet of Things (IOT), Vehicular ad-hoc Networks (VANETs) have become one of the hotspots in the research fields of intelligent transportation systems for scholars focusing on how to improve the efficiency and safety of the road [5, 11, 23, 25]. It is estimated that the number of registered vehicles around the world wil reach 2 billion within the next 10 to 20 years [4]. Based on the On-Board-Unit(OBU) installed on vehicle, VANETs include two types of communications:

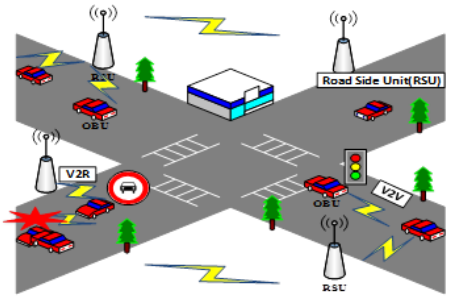


Figure 1: The architecture of VANETs

1) The Vehicle-to-Vehicle (V2V);

2) The Vehicle-to-Road Side Unit (V2R). The architecture of VANETs is shown in Figure 1. With the help of Road Side Units (RSUs), nearby vehicles can exchange traffic, weather and other information via the dedicated short range communication (DSRC) [10], which helps drivers make timely and reasonable driving strategies. In such situations, authentication and security need to be ensured.

However, due to the high mobility and variability of network topology, the system is vulnerable to be threaten by the malicious adversary in the VANETs. Therefore, security, privacy and authentication should be taken into account [2]. Specially, two types of issues, namely disclosure of location privacy and identity privacy pose some serious threats to the entire networks. Firstly, if the adversary

learns the location of a particular node, the node's communication behavior will be tracked and eavesdropped. In other words, an attacker can track the driving line of a user with a special identity. Secondly, it is extremely serious that malicious attackers launch a Sybil Attack by using these identity information stored in cloud servers. In order to provide secure communication environments, researchers used to focus on the traditional infrastructure, namely the public key infrastructure (PKI). Asymmetric cryptography algorithm and digital certificates are utilized in PKI, protecting identity information of the users via a centralized trusted third party (TA) [17]. However, with the number of vehicles increasing, the management of PKI certificates requires huge storage and computational overhead, especially for certificate revocation. Moreover, a single centralized trusted third party may cause a single point of failure. Therefore, how to provide an effective solution is still a problem remained to be solved urgently, such as efficiency and distributed security.

With all this in mind, blockchain is considered as a revolutionary technology to cope with the problems above. As the underlying technology of the Bitcoin, blockchain was initially proposed by Nakamoto in 2008 [16]. It utilizes a distributed database in the peer to peer (P2P) network to record all transaction behaviors and maintain a consistent and tamper-proof ledger. Due to high security and reliability, the combination of blockchain and VANETs has received considerable attention [21, 26]. On the one hand, in VANETs, all activities and information could be written into the immutable and unforgeable ledger, which can be verified and traced by all legitimate members. On the other hand, it can avoid single point of failure in a distributed way and enhance the security of the system.

1.1 Related Research

Compared with open access environment, providing a secure and reliable communication environment for vehicles plays an extremely important role in VANETs [3,27]. Therefore, authentication, privacy, and confidentiality of information should be taken into account seriously. Lin *et al.* [13] proposed a secure protocol based on group signature, which can guarantee privacy of users and provide the desired traceability for each vehicle. However, the pure group signature verification is usually time-consuming, and it is hard to meet the real-time requirements of the application in VANETs. For obtaining high privacy and security, Yao *et al.* [22] proposed a biometrics-based authentication scheme, which uses a temporary MAC address to conceal the real MAC address. Jiang *et al.* [6] adopt pseudonyms to realize batch authentication by using an identity-based signature (IBS). However, most of them are based on traditional digital signature technology of PKI, which have high computational and storage overhead. Vijayakumar *et al.* [20] proposed a secure authentication and key management mechanism to ensure the security of user's key in VANETs. Lim *et al.* [12] proposed an efficient protocol for fast dissemination of authentication messages, and Tan *et al.* [19] proposed a secure certificateless authentication to realize vehicle's identity authentication. However, these solutions rely on a centralized trusted third party and cannot provide the distributed security.

With the properties of decentralization, transparency, traceability and non-tampering, blockchain, a distributed public ledger shared and maintained by all nodes in the system, has attracted wide attention, not only in the financial industry but also in VANETs. Specifically, many researchers in VANETs focus on improving efficiency and security to ensure vehicle's privacy through blockchain technology. Yuan *et al.* [24] proposed a seven-layer conceptual model for Intelligent Transportation Systems (ITS) via blockchain technology, and claimed that the decentralized model will be the future of ITS. Dorri *et al.* [1] proposed a blockchain-based architecture to increase the security and protect the privacy of users. Although the privacy and security were considered in the paper, they do not give the concrete and practical scheme. Lei *et al.* [9] proposed a secure blockchain-based key management framework with a security managers (SMs) in ITS. Lu *et al.* [14] designed a decentralized anonymous reputation system using blockchain technology for VANETs. Rowan *et al.* [18] proposed a blockchain-based PKI and an inter-vehicle session key establishment protocol for secure V2V communications. In the above researches, the blockchain is applied to enhance security between information and energy interactions. However, these schemes are only suitable in Bitcoin. Malik *et al.* [15] proposed an authentication and revocation of framework using blockchain technology, which authenticates vehicles in a decentralized way. However, they store a certain number of bytes using the OP_RETURN instruction in Bitcoin. Actually, storing a large amount of non-transaction information in the Bitcoin network affects the performance of system, therefore, the size of OP_RETURN instruction is limited, and with the number of vehicles increasing, the amount of information stored in the blockchain will be enormous, which directly affects the scalability of the system.

1.2 Our Contributions

In this paper, we propose an anonymous authentication scheme based on consortium blockchain in VANETs, and the real identity of the vehicle can be concealed by using pseudo IDs to ensure the privacy of the vehicle. Specifically, we make the following contributions:

1) Our scheme allows the vehicles using Pseudo IDs obtained from the Road Side Units (RSUs) to conceal the real identity of the vehicle, which can prevent location tracking. Furthermore, each transaction includes a unique transaction ID (TID) and the RSU can quickly verify vehicular identity information using TIDs.

2) Based on consortium blockchain, we conduct a rigorous review of the nodes joining the system to ensure the confidentiality of the ledger and provide a decentralized, distributed, reliable database maintained by multiple trusted authorities (TAs) and RSUs for storing certificates. In addition, a great deal of anonymous certificates are stored in the trusted cloud server and pointers to the storage location are stored in the blockchain, which can improve the scalability of the system.

3) In the revocation, compared with searching the entire certificate revocation list (CRL), RSUs are able to determine promptly that the vehicle has been revoked by adding a revocation tag in our scheme, and the latter requires less computational overhead.

1.3 Organization

The remainder of this paper is as follows: Section 2 demonstrates a succinct concise overview of the consortium blockchain, VANETs and assumptions. In Section 3, the system model of anonymous authentication based on consortium blockchain for VANETS is discussed. In Section 4, the proposed scheme including registration, authentication and revocation is given. Section 5 analyzes the security of our scheme and evaluates the theoretical performance. Finally, Section 6 concludes the paper.

2 Preliminaries

2.1 Consortium Blockchain

The Consortium blockchain, a type of permission blockchain, is not completely decentralized, but it is a multicenter blockchain as shown in Figure 2. The predefined authoritative node A can select the accounting nodes 1, 2 and 3 by voting and the remaining nodes are the ordinary nodes. Compared with public blockchain, only legitimate nodes (member nodes) can access the ledger and view related information stored in consortium blockchain by setting up access permissions. In addition, the access authority and record authority of the ledger are determined jointly by authoritative nodes to ensure the confidentiality of ledger, and provide a higher security. Generally, considering the efficiency of the system, it do not use mining mechanisms, such as Proof of Work (POW) algorithm.

2.2 VANETs

VANETs, a special wireless ad-hoc network, can provide a secure and efficient environment for Vehicle-to-Vehicle (V2V) communication and Vehicle-to-Infrastructure (V2I) communication as shown in Figure 1. There are three types of entities in VANETs: Trusted Authority (TA), Roadside Unit (RSU), and On-Board Unit (OBU).

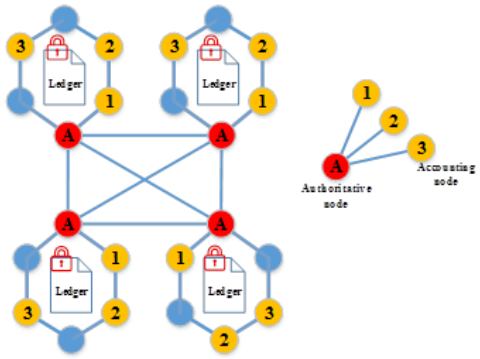


Figure 2: Structure of consortium blockchain

TA plays an extremely important role in the process of vehicle registration and authentication. Multitude of wireless gateway points, *i.e.*, RSUs, are deployed along the roadside. Through the RSU, Vehicles can share valuable driving information with neighboring vehicles to improve traffic efficiency and safety. OBU configured into vehicle is responsible for communicating with RSU by utilizing dedicated short range communication (DSRC) radio.

2.3 Assumption

The process of registration is divided into two phases. Firstly, the vehicle obtains the public key certificate from the TA. Secondly, it obtains the Pseudo ID from the RSU within the region covered by the TA. In addition, the distance affects the communication delay, therefore, the nearest RSU to the vehicle is responsible for generating PID in our scheme. All of the above are prerequisites for authentication, we need make the following assumptions:

1) We assume that TAs and RSUs generated Pseudo ID for vehicle are completely trusted and they are not be compromised.

2) When the vehicle obtains the public key certificate from the nearest TA, the locations of RSUs within the region covered by the TA are stored in the OBU installed in the vehicle. Therefore, at any time, the vehicle knows the nearest RSU and the RSU generates pseudo ID for vehicles quickly.

3) we assume that the cloud server in our work is absolutely trustworthy.

3 System Model

In this section, we introduce the system model and the specific function of each entity in the system model.

There are five entities in the proposed system: A Traffic Department (TD) , multiple Trust Authorities (TAs), Road Side Units (RSUs), On Board Units (OBUs) installed in vehicle and a Trusted Cloud Server (TCS). It is worth noting that TD, TA and RSU represent three types

of nodes, namely supervisory nodes, accounting nodes (revocation nodes) and verification nodes. As shown in Figure 3.

- Traffic Department: Firstly, as the supervisory node of the system, the traffic department is responsible for supervising the operation of the entire system. Secondly, the supervisory node needs to select the accounting nodes (TAs) in advance for generating the transaction information and uploading them to the blockchain. In addition, the vehicle needs to submit personal information to the TD before registration and obtain a unique plate number namely VID;

- Trust Authority: There are multiple authoritative nodes in our system and their accounting rights are granted by the TD. There are mainly three functions for TA. First, it is responsible for assigning a public-private key pair to the vehicle and RSU within the region coverd by the TA, which are used to authenticate between vehicle and RSU. Second, a candidate transaction set is generated by the TA including a large number of public key certificates encrypted using the public key of the TA. Finally, the TA uploads the integral transactions to the blockchain. It is worth noting that a integral transaction consists of a candidate transaction and a pointer to the storage location of pseudo ID;

- Road Side Unit: There are many RSUs distributed within the region covered by each TA. Each RSU is a verification node in the blockchain and is mainly responsible for generating a pseudo ID for the vehicles and sending the generated pseudo ID to the TCS. In addition, a pointer to the storage location of pseudo ID is transferred to the TA. RSU1 and RSU2 represent two different RSUs;

- On Board Unit: Due to the limited resources and computing power of OBU, it only participates in the simple encryption and transmission of data, and sends the collected data as a data set to the RSU;

- Trusted Cloud Server: We can only upload the user's real identity and the hash index of the pseudo ID to the blockchain, and a large number of pseudo IDs are sent to a Trusted cloud server. Here, we assume that this cloud server is absolutely trustworthy.

4 The Proposed Anonymous Authentication Scheme in VANETs

In this section, we describe the blockchain-based anonymous authentication scheme in detail including system initialization, registration, mutual authentication and expeditious revocation.

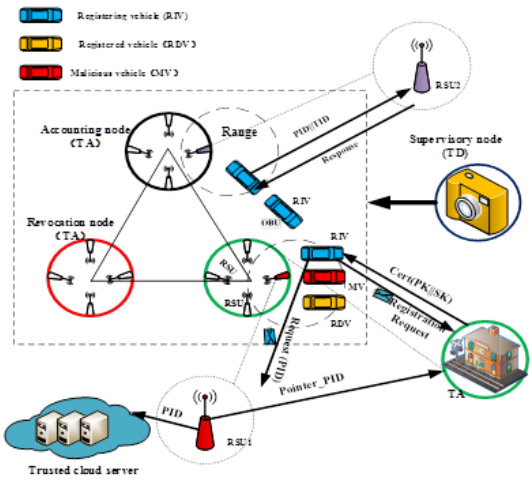


Figure 3: System model

4.1 System Initialization

The notations used in this paper are given in Table 1.

Table 1: Notations

Notation	Meaning
P_{V_i}	The public key of vehicle
SK_{V_i}	The private key of vehicle
P_{R_i}	The public key of i^{th} RSU
$TX_i()$	Candidate transaction set
$T_i()$	Timestamp
$TID_j()$	Transaction ID of j^{th} transaction
$POINTER$	A pointer to the storage location of Pseudo ID
$E()$	Encrypt
$Sig()$	Digital signature
R	Random numbers

The system is comprised of five participants: Traffic Department (TD), multiple Trust Authorities $TA = \{TA_1, TA_2, ..., TA_n\}$, vehicle sets $V = (V_1, V_2, ..., V_i)$, Roadside Units $R = \{RSU_1, RSU_2, ..., RSU_i\}$ and Trusted Cloud Server. In the registration, different participants prepare to be occupied with numerous domain parameters required for security operations. The system is maintained by multiple Trust Authorities for Elliptic curve cryptography (ECC) based PKI technology, and system parameters set $\{q, a, b, P\}$ is initialized. Here, a and b are constants defining the Elliptic curve equation ($a, b \in F_q$ and $4a^3 + 27b^2 \neq 0$). P is the generator of the Elliptic Curve E with prime order q. There are many RSUs within coverage of each TA. We assume that $TA_1 \in TA$ needs to distribute ECC public-private key pairs to the RSUs. TA_1, one of multiple trusted authorities, selects a integer set $(a_1, a_2, ..., a_n \in Z_q)$ as private keys of RSUs and generates a public key set $(P_{R1}, P_{R2}, ..., P_{Rn})$, where $P_{Rn} = a_i \cdot P$.

The consortium blockchain is established amoung TD, multiple Trust Authorities and RSUs. Multiple Trust Authorities are responsible for generating new identities for vehicles. The TA_1 elected as a accounting node by using suitable voting mechanism uploads transaction information to the blockchain.

4.2 Registration of the Vehicle

In our work, there are two stages for the vehicle to complete the registration. Firstly, the TA is responsible for generating an ECC public-private key pair namely P_{V_i} and SK_{V_i} for the vehicle, and generating a candidate transaction set waiting for being uploaded. Secondly, the RSU generates a pseudo ID for the vehicle.

1) The TA generates a Public-Private key pairs for V_i:

Table 2: Registration of the vehicle

$1.V_i \rightarrow TA_1 :< VID_i \| \| Other >$
$2.TA_1 \rightarrow VID_i :< Verify(VID_i) >$
$3.TA_1 \rightarrow V_i :< d \| \| P_{V_i} \| \| Sig\{H(P_{V_i} \| \| d)\} \| \| T_1 >$
$4.TA_1 \rightarrow V_i :$ $< TX_i\{E(Cert_{V_i})\} \| \| TX_{i+1}\{E(Cert_{V_{i+1}})\} >$

The steps of registration are described in Table 2. The vehicles register with TA for the first time by submitting their VID_i issued by TD. The supervisory node (TD) in the system need to select a node being responsible for the registration of the vehicle according to a specific consensus algorithm. Here, we assume that TA_1 is only an authoritative node that is temporarily elected for this registration.

TA_1 verifies the VID_i and selects a integer $b \in Z_q$ as the private key of the vehicle namely $SK_{v_i} = b$ and generates a public key P_{V_i}, where $P_{V_i} = b \cdot P$. TA_1 send $< b, P_{V_i}, H(Sig), Sig, T_1 >$ to the vehicle through a secure channel, and at the same time, it generates a partial transaction set waiting for being uploaded including real identities of a large number of vehicles.

It is worth noting that the public key certificates are stored in the partial transaction set in the form of ciphertext, and they are encrypted with the public key of the TA_1.

2) The RSU generates a Pseudo ID for V_i:

The vehicle sends a request message encrypted with public key of RSU_1 including the public key certificate P_{V_i} obtained from the TA_1 and timestamp T_1. After receiving the request message, RSU_1 can select a random number $R_1 \in Z_q$ and calculate a message $M_1 = a \cdot P_{V_1} \cdot R_1$ for the vehicle. The vehicle selects a random number $R_2 \in Z_q$ and calculates two messages M_2, M_3, where $M_2 = b \cdot P_{R_1} \cdot R_1$ and $M_3 = R_2 \cdot M_1$. Here, (P_{R_1}, a) is the public-private key pair of RSU_1. The vehicle sends a reply message $M = M_2 || M_3$ and a T_1 to RSU_1, and RSU_1 can verify the identity of the vehicle by determining if M_3 is equal to M_4, where $M_4 = M_2 \cdot R_2$. After the identity of the vehicle V_i is authenticated, the RSU sends the pseudo ID with the timestamp T_1 to the vehicle and at the same time, the vehicle has completed registration.

Algorithm 1 Generation of Pseudo ID

```
Begin
A vehicle V_i wants to send a Request to the nearby RSU_1.
Let Request =< E_{P_{R_1}}(Cert_{V_i}(P_{V_i})||T_1) >.
The RSU_1 receives the Request from V_i.
Let M_1 = a · P_{V_1} · R_1.
The RSU_1 sends to the M_1 to V_i.
The vehicle V_i sends to a Reply to the nearby RSU_1
Let M_3 = R_2 · M_1, and M_2 = b · P_{R_1} · R_1.
Let Reply = M_3||M_2.
The RSU_1 verifies the information of the vehicle V_i.
Let M_4 = M_2 · R_1.
if M_3 = M_4 then
    Let M = PID_i||T_1
    Send message M to the vehicle.
end if
Periodically refresh the PID_i
End
```

4.3 Uploading Transaction to Blockchain

After sending the pseudo ID to vehicle, the RSU_1 will send the PID_i generated for this vehicle to the Trusted Cloud Server and forward a pointer to the memory address of PID_i namely $POINTER\_PID_i$ to the TA_1.

The TA_1 records the pointer in the partial transaction previously waiting to be uploaded. At the same time, the TA_1 generates a complete transaction set and uploads it to the blockchain. In addition, we redefine contents of each transaction in blockchain, and each transaction includes a public key certificate encrypted by using public key of T_1, a pointer and a transaction ID as shown in Figure 4. The registration information of the vehicle forms a transaction with a uniquely identified transaction ID, namely TID_j. Using transaction ID, we can determine the identity of a vehicle by viewing records stored in the blockchain.

4.4 Mutual Authentication Between RSU2 and Vehicle

The vehicle V_i leaves the region covered by RSU_1 and enters a region covered by RSU_2 as illustrated in Figure 5. It is critical for the vehicle and RSU_2 to complete

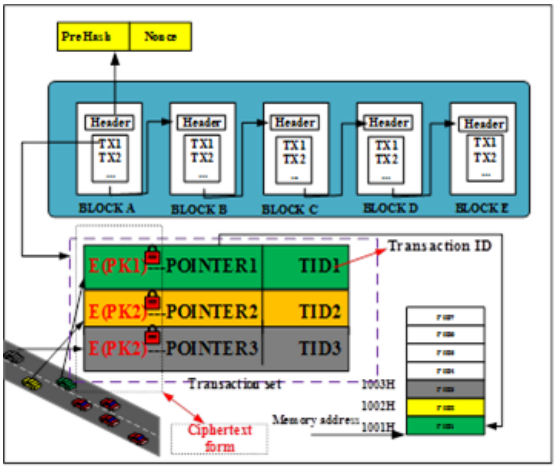


Figure 4: Transaction format of our scheme

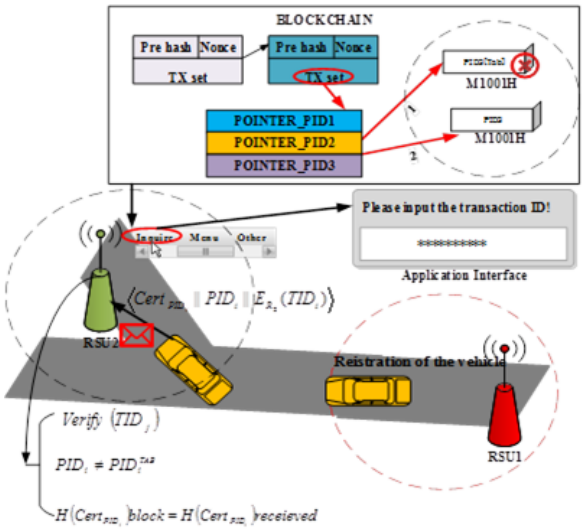


Figure 5: Mutual authentication between RSU_2 and vehicle

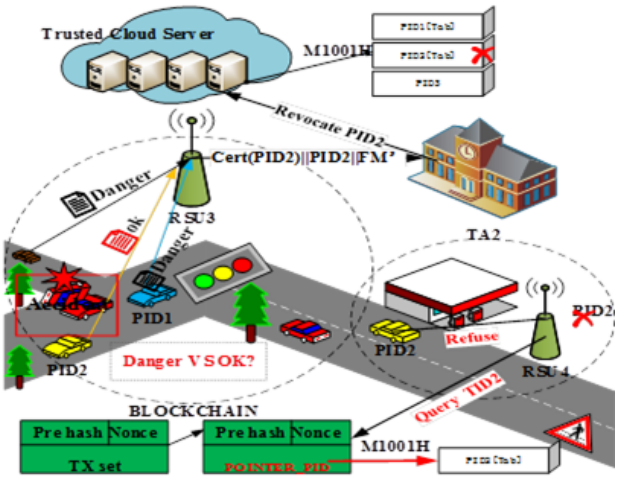


Figure 6: Revocation of malicious vehicle

anonymous authentication. The authentication process is divided into five steps:

- **Step 1:** The vehicle sends an authentication message M_i including PID_i, $Cert_{PID_i}$, $H(Cert_{PID_i})$, a timestamp T_2 and a transaction ID encrypted with the public key of RSU_2 namely $E_{P_{R_2}} < TID_j >$ to RSU_2.
- **Step 2:** After receiving the message M_i, the RSU_2 decrypts the message M_i by using its private key a_2 and gets the PID_i transaction ID (TID_j), and timestamp T_2. The RSU_2 can verify the legality of the vehicle by querying the blockchain using TID_j.
- **Step 3:** Based on the transaction ID provided by the TA_1, the RSU_2 can quickly know identity information of the vehicle by visiting transaction information instead of traversing the entire blockchain system.
- **Step 4:** Firstly, through the transaction information recorded in the blockchain, the RSU_2 determines whether the transaction information corresponding to the TID_j exists. If it does not exist, the vehicle can be considered as an illegal node. Secondly, if a pointer to PID_i has a revocation tag namely PID_i^{TAB}, the information provided by the vehicle is invalid. Finally, RSU_2 can verify whether the message has been tampered with by comparing the $H(Cert_{PID_{receieved}})$ with $H(Cert_{PID_i})$. If the equation $H(Cert_{PID_i}) = H(Cert_{PID_{receieved}})$, the vehicle is legal.
- **Step 5:** Once the legality of vehicle identity is verified, RSU_2 can provide the corresponding service to it.

4.5 Expeditious Revocation

In the revocation, we assume that there are some reports: "Dangerous", "OK" and "dangerous" from three vehicles in the region covered by RSU_3 for the same road condition, and contents of the message are proven fallacious by using the evaluation algorithm. As shown in Figure 6, PID_1, PID_2, PID_3 represent three different vehicles respectively.

When the RSU finds that the PID_2 is sending "Forged Message" ("FM"), the RSU forwads a message including $Cert_{PID_2}$, PID_2 and "FM" to TA_2, and the message is encrypted by using the public key of TA_2. Once verified, the TA_2 sends a revocation command to the Trusted Cloud Server (TCS) through a secure channel. In our work, we set a revocation tab PID_2^{TAB}. The TA_2 is responsible for updating ledger in this paper. When the vehicle (PID_2) enters the region covered by the RSU_4, the RSU_4 can query the information stored in the blockchain and determine whether the vehicle has been revoked. Because of obtaining the PID_2^{TAB} instead of the (PID_2), the system refuses to provide the corresponding service for vehicles.

The data stored in the blockchain is just a pointer to the storage location. When a malicious vehicle is found, the information of vehicle can be modified without changing the transaction itself. In addition, compared with searching the complete revocation list(CRL), we just need to determine whether the content of the pointer is PID_i^{TAB} that requires lower computational overhead.

5 Security and Performance Analysis

5.1 Security Analysis

- **Confidentiality:** In the registration, the vehicle calculates message $M_1 = a \cdot P_{V_i} \cdot R_1$, where a is the private key of vehicle. RSU_1 calculates messages $M_2 = b \cdot P_{R_1} \cdot R_1$ and $M_3 = R_1 \cdot M_1$, where b is the private key of RSU_1. Two parties of the communication complete the mutual authentication by determining whether $M_2 \cdot R_1$ is equal to $R_2 \cdot M_1$. Messages encrypted with their public key can't be decoded, unless the attacker can obtain their private key. Specifically, the process of obtaining the private key is an ECDLP problem. Therefore, our scheme satisfies confidentiality.

- **Anonymity:** In the mutual authentication between the vehicle and the RSU_2, the vehicle sends a message M including PID_i, T_2, $Cert_{PID_i}$ and TID_i to the RSU_2, namely $< E_{P_{R_2}}(PID_i||Cert_{PID_i}||T_2||TID_i) >$. The RSU_2 decrypts it by using its private key, and determines whether the equation $H(Cert_{PID_i}) = H(Cert_{PID_{receieved}})$ is true by querying the information stored in the blockchain. In the authentication, the real identity of the vehicle can be concealed by using PID_i, which can ensure the anonymity of the vehicle.

- **Single point of failure:** There is no single point of failure in our scheme. Firstly, multiple Trusted Authorities (TAs) and RSUs jointly maintain a reliable ledger with authority. Each TA is responsible for distributing public-private key pairs for vehicles and RSUs. Secondly, in order to weaken permissions of the authoritative node TA, the RSU generates a pseudo ID for the vehicle in our scheme. Ultilizing a blockchain with authority can ensure distributed features. In addition, we have restricted on access to the ledger, so not all nodes can view the information stored in the blockchain.

- **Unforgeability:** Attackers generally complete authentication by forging the user's identity. We assume that the attacker forges the identity of the vehicle and calculates $M_2' = c \cdot P_{R_1} \cdot R_1$, where c is the private key of attacker. In our work, the vehicle calculates message $M_2 = b \cdot P_{R_1} \cdot R_1$ and sends it to RSU_1. The equation M_2' is not equal to M_2, unless the attacker can obtain the private key of the vehicle. The equation $M_2' \cdot R_1 \neq R_2 \cdot a \cdot P_{V_1} \cdot R_1$, the RSU failed to verify the identity of vehicle that the registration was unsuccessful.

- **Repaly attack:** The attacker achieves the purpose of deceiving the system by sending the same packets repeatedly. However, The process of authentication is based on random numbers $R1$, $R2$, and the random number can only be known by itself. It can ensure that there is no fixed connection for the request and reply between the vehicle and RSU, so the vehicle's private key cannot be decoded by the replay attack.

5.2 Performance Analysis

In this section, we analyze the feasibility of our scheme in terms of time consumption, storage capacity and security. The scalar multiplication operation on the Elliptic Curve, encryption operation, decryption operation and hash operation will be involved. In addition, it also involves digital signature and verifying. In our paper, we use the Elliptic Curves recommended by [7] and all operations are based on the ECC algorithm. Specially, we refer to the time of the scalar multiplication operation used in [8]. For the convenience of description, it will be defined in Table 3.

Table 3: The operation involved in this scheme

Operation	Time
T_{mul}	The time of a scalar multiplication operation
T_{sig}	The time of one digital signature
T_{Veri}	The time of verifying the signature
T_H	The time of hash operation
T_{enc}	The time of encryption operation
T_{dec}	The time of decryption operation

Specifically, our scheme involves digital signature, verifying, encryption operation, decryption operation and four scalar multiplication operations in the registration. The computation overhead of a vehicle can be summarized as: $4T_{mul} + 1T_{sig} + 1_{Veri} + 1T_{dec}$. There are multiple trusted authorities (TAs) in this paper. We assume that the maximum number of vehicles supported by a TA is 100 and n represents the number of vehicles. Under different values of n, the time consumption is tested in the registration and the authentication. As shown in Figure 7, in the registration, the total time taken for the 20 vehicles to complete the registration is 644.712ms. The number of vehicles increased from 20 to 100, and the total time taken is 3235.193 ms, which is the maximum time spent on registration.

In the authentication, hash operation, encryption operation and decryption operation based on ECC will be

Table 4: Comparison of security and function

Scheme	Anonymity	Decentralization	Tamper-Resistant	System Scalability
[13]	√			
[22]	√			
[6]	√			
[15]	√	√	√	
Our Scheme	√	√	√	√

involved, the computation overhead can be expressed as: $1T_H + 1T_{enc} + 1T_{dec}$. In authenticating, vehicles provide $H(Cert_{PID_i})$ to the RSU, and the RSU can authenticate legality of the vehicle by comparing with $H(Cert_{PID_i})_{block}$. Determining whether the message has been tampered with, we need to search its PID information and perform a hash operation. As shown in Figure 8, The time taken for 20 vehicles to complete the authentication is 5.245ms, The number of vehicles increased from 20 to 100, and the total time taken is 36.79ms.

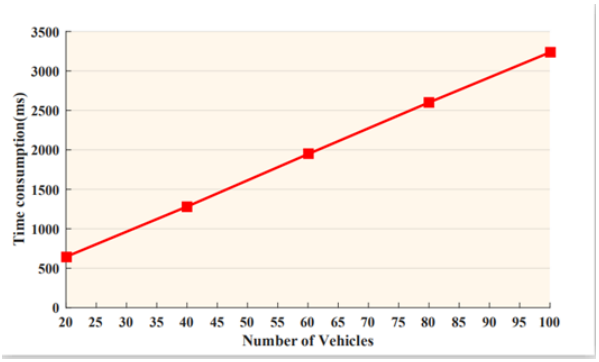


Figure 7: Registration of vehicle

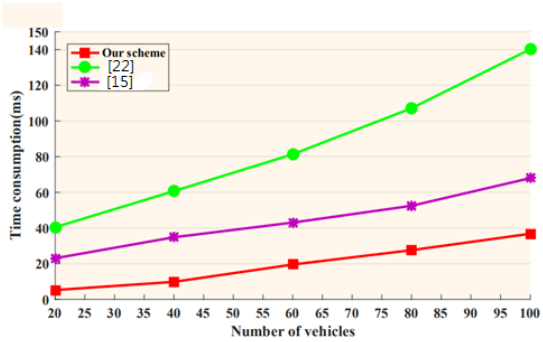


Figure 8: Authentication of vehicle

Figure 8 compares the time consumption of our scheme with [15, 22] under different values of n. Yao *et al.* [22] proposed an anonymous authentication scheme, seven encryption operations, six hash operations are required in their scheme. The scheme of [15] requires three encryption operations, two decryption operations, one hash operation. Compared with our scheme, their scheme authenticating 20 vehicles takes 40.454ms and 23.052ms respectively. In addition, the maximum time spent on authentication is 140.216ms in [22]. The results of simulation demonstrate that our proposal can meet the real-time performance of the VANETs.

For revocation, different from searching the complete certificate revocation list (CRL) bringing huge computational overhead, we introduce a revocation tab. Once the system considers that the vehicle is a malicious node, the PIDs stored on the trusted server will be marked with a tab. According to the blockchain, the RSU can determine whether the vehicle has been revoked by obtaining a PID_i instead of PID_i^{TAB}.

In VANETs, many authentication schemes are based on the Bitcoin system. For example, in [15], they only store a certain number of bytes using the OP_RETURN instruction in bitcoin. In the Bitcoin system, Bitcoin developers believe that OP_RETURN will cause users to store too much non-transaction information in the Bitcoin network affecting the system performance of Bitcoin, therefore, the storage space is strictly restricted. However, with the number of vehicles increasing, the number of information stored in the blockchain will be enormous, which will directly affect the scalability of the system. In our scheme, only the pointer to PID_i are stored in the blockchain, and the PID is stored in the trusted cloud server. The storage capacity of the trusted server is undoubtedly huge. Therefore, we do not worry about the storage problems caused by the explosion of vehicles.

In addition, as shown in Table 4, we compare it with schemes [6, 13, 15, 22] in terms of anonymity, tamper-resistant and decentralization. Our scheme has more advantages in security and function.

6 Conclusions

Aiming at providing a distributed security, in this paper, we propose an authentication scheme based on consortium blockchain with anonymous identity in VANETs. The anonymity of vehicles can be guaranteed by using PIDs to conceal the real identity of users. In order to improve the scalability of the system, we introduce a trusted cloud server to store the PIDs, and location pointers are uploaded to the blockchain. In addition, a vehicle can be considered an illegal node by judging whether the PID has a revocation tab instead of searching the entire certificate

revocation list (CRL). Finally, we analyze the security of our scheme and evaluate the performance of the anonymous authentication scheme.

Acknowledgments

This work was supported by the Natural Science Foundation of China under Grants 61802303 and 61772418, the Innovation Ability Support Program in Shaanxi Province of China under Grant 2017KJXX-47, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2016JM6033 and 2018JZ6001.

References

[1] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.

[2] F. Dötzer, "Privacy issues in vehicular ad hoc networks," in *International Workshop on Privacy Enhancing Technologies*, pp. 197–209, May 2005.

[3] K. K. Gai, M. K. Qiu, Z. G. Xiong, and M. Q. Liu, "Privacy-preserving multi-channel communication in edge-of-things," *Future Generation Computer Systems*, vol. 85, pp. 190–200, 2018.

[4] D. Y. Jia, K. J. Lu, J. P. Wang, X. Zhang, and X. M. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 263–284, 2016.

[5] D. Y. Jia and D. Ngoduy, "Enhanced cooperative car-following traffic model with the combination of V2V and V2I communication," *Transportation Research Part B: Methodological*, vol. 90, pp. 172–191, 2016.

[6] S. R. Jiang, X. Y. Zhu, and L. M. Wang, "An efficient anonymous batch authentication scheme based on HMAC for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2193–2204, 2016.

[7] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

[8] H. H. Kilinc and T. Yanik, "A survey of sip authentication and key agreement schemes," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1005–1023, 2014.

[9] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. Ogah, and Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, 2017.

[10] Y. J. Li, "An overview of the DSRC/WAVE technology," in *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pp. 544–558, Nov. 2010.

[11] C. L. Li, Y. Zhang, T. H. Luan, and Y. C. Fu, "Building transmission backbone for highway vehicular networks: Framework and analysis," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8709–8722, 2018.

[12] K. Lim and D. Manivannan, "An efficient protocol for authenticated and secure message delivery in vehicular ad hoc networks," *Vehicular Communications*, vol. 4, pp. 30–37, 2016.

[13] X. D. Lin, X. T. Sun, P. H. Ho, and X. M. Shen, "Gsis: A secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007.

[14] Z. J. Lu, W. C. Liu, Q. Wang, G. Qu, and Z. L. Liu, "A privacy-preserving trust model based on blockchain for VANETs," *IEEE Access*, vol. 6, pp. 45655–45664, 2018.

[15] N. Malik, P. Nanda, A. Arora, X. J. He, and D. Puthal, "Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks," in *The 17th IEEE International Conference on Trust, Security And Privacy in Computing And Communications/12th IEEE International Conference on Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 674–679, 2018.

[16] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. (`https://bitcoin.org/bitcoin.pdf`)

[17] A. Nash, W. Duane, C. Joseph, D. Brink, *PKI: Implementing and Managing E-security*, 2001. ISBN 13: 978-0072131239.

[18] S. Rowan, M. Clear, M. Gerla, M. Huggard, C. M. Goldrick, "Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels," *arXiv Preprint arXiv:1704.02553*, 2017. (`https://arxiv.org/pdf/1704.02553.pdf`)

[19] H. Tan, D. Choi, P. Kim, S. Pan, and I. Chung, "Secure certificateless authentication and road message dissemination protocol in VANETs," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 13, 2018.

[20] P. Vijayakumar, M. Azees, A. Kannan, and L. J. Deborah, "Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1015–1028, 2015.

[21] Z. Yang, K. Yang, L. Lei, K. Zheng, and Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2018.

[22] L. Yao, C. Lin, G. W. Wu, T. Y. Jung, and K. B. Yim, "An anonymous authentication scheme in data-link layer for VANETs," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 22, no. 1, pp. 1–13, 2016.

[23] M. B. Younes, "Secure traffic efficiency control protocol for downtown vehicular networks," *International Journal Network Security*, vol. 21, no. 3, pp. 511–521, 2019.

[24] Y. Yuan and F. Y. Wang, "Towards blockchain-based intelligent transportation systems," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC'16)*, pp. 2663–2668, Nov. 2016.

[25] T. Zhang and Q. Y. Zhu, "Distributed privacy-preserving collaborative intrusion detection systems for VANETs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 148–161, 2018.

[26] D. Zheng, C. M. Jing, R. Guo, S. Y. Gao, and L. Wang, "A traceable blockchain-based access authentication system with privacy preservation in VANETs," *IEEE Access*, vol. 7, pp. 117716–117726, 2019.

[27] L. Y. Zhu, C. Chen, X. Wang, and A. O. Lim, "Smss: Symmetric-masquerade security scheme for VANETs," in *Tenth International Symposium on Autonomous Decentralized Systems*, pp. 617–622, Mar. 2011.

Biography

Liang Wang received the B.S. degree from the Institute of Information Technology, GUET, in 2016. He is currently pursuing the M.S. degree with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, China. His research interests include anonymous authentication, vehicular ad hoc networks, and blockchain technology.

Dong Zheng received the Ph.D. degree from Xidian University, in 1999. He joined the School of Information Security Engineering, Shanghai Jiao Tong University. He is currently a Professor with the Xi'an University of Posts and Telecommunications, China. His research interests include information theory, cryptography, and information security. He is also a Senior Member of the Chinese Association for Cryptologic Research and a member of the Chinese Communication Society.

Rui Guo received the Ph.D. degree from the State Key Laboratory of Networking and Switch Technology, Beijing University of Posts and Telecommunications, China, in 2014. He is currently a Lecturer with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications. His current research interests include attribute-based cryptograph, cloud computing, and blockchain technology.

ChenCheng Hu received the B.Eng. degree from the Xi'an University of Posts and Telecommunications, in 2016, where he is currently pursuing the M.S. degree. He is doing research at the National Engineering Laboratory for Wireless Security. His current research interests include blockchain technology, user authentication, and information security.

ChunMing Jing received the bachelor's degree from the Xi'an University of Posts and Telecommunications, in 2017. He is currently pursuing the master's degree with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications. His research interests include blockchain technology, vehicular ad hoc networks, and security and privacy in the Internet of Things.

A Literature Survey of Visual Similarity Snooping Attacks in Emails

George Mwangi Muhindi, Georey MarigaWambugu, and Aaron Mogeni Oirere
(Corresponding author:George Mwangi Muhindi)

School of Computing and Information Technology, Murang'a University of Technology, Kenya
Murang'a University of Technology, P.O.Box 75-10200, Murang'a, Kenya
(Email: georgemuhindi@gmail.com)
(Received May 13, 2020; revised and accepted July 5, 2020)

Abstract

Snooping is one of the most significant issues that the cybersecurity industry faces in this modern era of technology, leading to substantial financial losses for individuals and organizations. The detection of snooping attacks with efficiency and preciseness is proving to be a challenge due to the complex nature of the snooping attacks. A snooping website appears to be very similar to the corresponding genuine website, which deceives the unknowing users to believe that they are on the right site. Banks and other financial institutions should prevent loss of money through snooping attacks. To achieve this, they should understand how the snooping attacks occur and the techniques that can be used to detect visual similarity snooping attacks. There is also a need to develop and implement a mechanism that can check against snooping attacks, and this can be achieved by checking for malicious links and attachments.

Keywords: Email; Malware; Snooping Attacks

1 Introduction

The email security threat has risen to become one of the biggest threats to companies across the world. Subsequently, it can be noted that a majority of the hacking attacks begin with some form of snooping attack. Snooping can be described as a kind of attack which is engineered socially to steal private and confidential data like passwords, credit card information, and login details. These snooping attacks occur when the attackers masquerade as genuine and trusted entities and end up tricking the unknowing users into opening the spammed emails [2]. When the recipient of the email receives the email and clicks on the embedded emails, malicious malware from the links infiltrate the computer system and, in the process, end up accessing and stealing private, sensitive, and confidential information.

The fake emails typically look very legit and genuine, and even the links that the user is asked to click on appear to be very legit when they request personal information. The snooping messages propagate themselves past instant messages, social media sites, emails, and VoIP. Nonetheless, email is the most popular way of carrying out the snooping attacks. It is true since 65% of the snooping attacks occur when the user clicks on a link and visits the hyperlink attached in the snooped email. More complicated snooping attacks target specific persons or groups from a firm [17]. Metaphorically, snooping is the same as fishing in a lake; rather than attempting to fish a fish, the attackers try to steal the user's personal information [27]. When the user unknowingly opens the fake website and feeds personal information such as login details, these personal details are acquired by the hacker who can then use this information for other malicious intentions.

The snooping websites have an appearance that is very similar to the genuine website to attract many users to the website [24]. With the development of snooping detection techniques, new approaches have been developed to detect visual similarity attacks. Optical similarity-based techniques use comparisons of the suspicious websites' visual appearance in correspondence to the genuine website by analyzing different parameters.

Banks should ensure that they prevent financial losses due to snooping attacks and should come up with techniques for preventing more snooping attacks. Moreover, there should be a technique that should check for snooping attacks before they occur. It should be done by checking for malicious links and attachments in the snooped emails sent to unsuspecting victims [22, 24].

2 Background and Statistics of Snooping Attacks

Snooping attacks and scams have gained both corporate and academic scholars' attention since this issue has led to serious privacy breaches and adverse security issues in the banking industry, resulting in the loss of millions of dollars. Snooping attacks cannot be mitigated through

the use of encryption software and firewalls.

The first snooping attacks that were experienced took place on the American online network systems (AOL), which occurred during the onset of the 1990s. There were many fraudulent users registered on the AOL site using fake credentials. The AOL verified fake accounts using a simple test without analyzing the validity of the credit cards. After activating the fake accounts, the hackers could access the different resources offered by the American online system. During the billing process, AOL was able to find out that the charges were illegitimate, together with the fact that the linked credit cards were not valid [2]. Thus, AOL stopped and closed down the accounts with immediate effect. After this incident, the American online networks system put measures that would ensure that the same does not happen in the future. The AOL put in place measures to prevent this by verifying and authenticating the credit cards linked to the billing accounts. It also enabled the attackers to switch, making it possible for them to obtain the AOL accounts. Rather than now creating the fake accounts, they changed to stealing the personal data of the users that were registered on the AOL system. The attackers then contacted the registered users using emails and instant messages, requesting them to verify their personal information and passwords for security reasons [10]. The emails and the news appeared to be originating from the AOL employees. It ended up duping most users to provide their personal information and passwords to hackers. The attackers, in turn, used confidential information in place of valid customers. Here, the attackers did not restrict themselves to masquerading as the actual AOL users but also, they actively tricked many other commercial websites in the USA.

As per research by the Internet World Stats, the total number of users on the internet stood is 2.97 billion in 2014, and by 2019 this number stood is 4.39 billion. This number is expected to rise as the years keep ongoing. With these figures in mind, over 38% of the global population makes use of the internet [26]. Many internet users give hackers the chance to take advantage of unknowing users and insecure online systems to scam users. Snooping emails are used for defrauding people and financial firms of money using the internet [8].

In 2012, there was a general increase in the number of snooping attacks translating to a 160% increase from the previous year. The total number of snooping attacks detected in the year 2013 stood at close to 45000, resulting in financial losses that stood at over 5.9 billion dollars. It meant a 1% increase in the number of snooping attacks from 2012 to 2013. The total number of snooping attacks observed in the first quarter of the year 2014 stood at 125215, and this was a 10.7% rise from the fourth quarter of 2013. Over 55% of the snooping sites have a similar name to that of the target website to dupe the user. Research has shown that the financial industry's snooping attackers mostly target payment systems and services [18].

3 The Mechanism of Snooping Attacks

The snooping mechanism is shown in Figure 1. The fake site is the clone of the simple website that the hackers target. It always has input fields such as the text area where the targeted user enters their personal information, then transferred to the hacker [3]. The hacker then steals this personal information from the unknowing user through the following steps:

Developing the snooping website: It is the initial step that the hacker takes by identifying the target or the organization. The attacker then gathers comprehensive information about the company by vising the website of the organization. The attacker then uses the data to develop a similar website [1].

Sending the URL: Here the hacker creates an email that is bogus and sends to many users. In the email, the hacker has attached the URL of the fake website. The attacker can also spread the snooping site's link using blogs or social media sites to reach many users [29].

Stealing the confidential information: When the unknowing user clicks on the embedded link, the fake website opens in the browser. The phony site has a fake login interface or login form that the attacker uses to steal personal information from the victim. Moreover, the attacker can gain access to confidential information that the user has filled up [13].

Identity theft: The hacker then uses the personal information obtained for malicious purposes. For instance, the hacker may make purchases online using the credit card information of the unknowing victim [15].

4 The Taxonomy of Visual Similarity Snooping Attacks

The attacker carries out the snooping attack by using social engineering mechanisms and technical subterfuge. With social engineering mechanisms, hackers manage to attack unknowing users by sending out bogus emails to thousands of unsuspecting users. The attackers typically convince the recipients of the emails to respond to the emails by keying their names, bank details, credit card firms, and e-retailers [15]. The technical subterfuge mechanism installs malware into the user's computer system. In the process, personal and confidential information is stolen by the use of crucial logger spyware and Trojan malware. The malware also misdirects users to websites that are fake or proxy servers [32]. The hackers embed malicious links or fraudulent links/ URLs in the emails

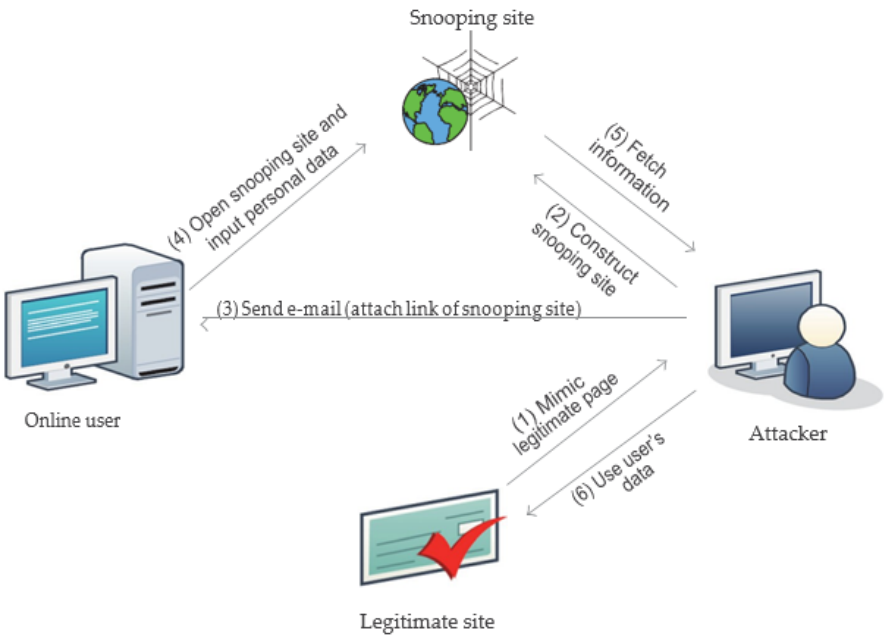


Figure 1: The snooping mechanism

that install malicious applications or software in the user's system. The malicious software then collects confidential data from the system and sends it back to the hackers. The hackers can also remotely access the user's computer system and then gather the data that they deem necessary [15].

A person can quickly become a snooping attack victim due to the visual similarity snooping site's high visual resemblance with the simple site because of the page set up, image layouts, font color and size, and the content. Figure 2 is an example of a fake and a genuine messenger of PayPal. The websites have the same visual appearance; however, one can observe that the URLs are different on a keen look. People are not always careful to take note of the URL and the SSL Certificates of the sites [7, 31].

If the hacker does not manage to copy the visual resemblance of the website being targeted, then the probability of the users inputting their credentials is minimal [19]. The hacker aims to fool the users using the following ways:

Through visual appearance: The snooping website has a similar look to that of the authentic website. The hackers steal a copy of the source code to build a legitimate website to develop a fake website.

Address bar: The hackers also hide the URL or the address bar of the site using an image or a script. It makes the users think that they are keying information on the legit site.

Embedded objects: The hackers also utilize embedded objects, such as scripts and images, to conceal the HTML code or the textual content from the snooping detection mechanisms.

Favicon similarity: It refers to an image that is linked to a specific site. A hacker can copy the image of the website that is targeted. If the shown favicon in the address differs from the current website, it is regarded as a snooping attempt.

Research conducted by Dhamija *et al.*, on different users to Identity if a website is genuine or a snooping site established that 90% of the users could not recognize snooping attacks [11]. A majority of the users judged the website wrongly using its visual appearance. The scholars also found out that even the experienced users could easily be duped through the illegitimate website's visual similarity. They also noted that 23% of the participants do not take time to view the site's address bar. Thus, it can be concluded that if the appearance of the snooping website looks exactly like that of a legit website and with a different domain, then the users can easily be fooled by the snooping attackers [4].

5 Mechanisms for Detecting Snooping Attacks

The following are mechanisms that have been devised to detect snooping attacks: Attribute-based, Identity-based, Content-based, and Character-based.

1) **Attribute based anti-snooping technique:** The attributes based anti-snooping technique executes every proactive and-anti-snooping technique defenses. This method has also been reinforced in Phish Bouncer Tool [21]. A comparison of images

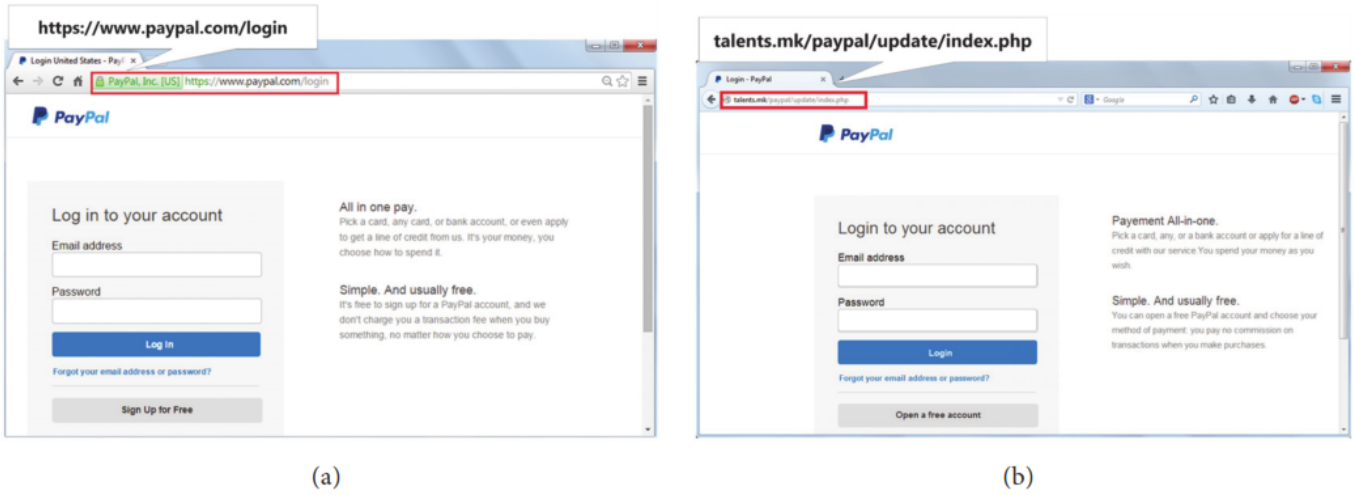


Figure 2: (a) Genuine PayPal webpage and (b) Snooping webpage of PayPal

visiting the website will be done using the image attribution check and checking for sites that are already registered under the Phish chucker-out. The HTML cross-link checks for responses that originate from websites that are not registered and count the different links that are not from registered websites [9]. When there is a large number of cross-links, it shows there is a snooping site. In the feeder check of false information, the false data is keyed in, and if the website accepts this information, then there is a high chance the link is also snooped. The anti-snooping suspicious check analyzes and validates the certificates given throughout the SSL handclasp. It carries on to daily usage by logging in for certification authority as time goes by [9].

Pros: This technique takes into consideration a lot of checks so that it can identify snooping websites in comparison to the other methods. The method can also detect snooping attacks that are known and those that are not known [25].

Cons: Because the technique carries out many checks for authentication of a website, there is a high probability of slow response time [9].

2) **Identity based anti-snooping techniques:** This mechanism makes use of the methodology of mutual authentication where an online entity and every user confirms each other's Identity through test suggestibility or handclasp. The method is associated with the technique of nursing ant snooping, which uses partial credential sharing alongside shopper filtering mechanism to hinder the attackers from pretending to be legit online users [16]. Mutual authentication is followed in this method; hence there is no need for the users to re-enter their details. Therefore, using passwords has never changed between the users and online entities, except the first method of setting it up [9].

Pros: This technique provides for mutual authentication for the client and server-side. Making use of this technique does not expose the personal details of a user, for example, the password that is set up except for the initial time that it is set up [5].

Cons: Using this technique, if the attacker can access the user's computer and then disable the browser plugins, it ends up being compromised [9].

3) **Content based anti-snooping approaches:** The GoldPhish tool executes this technique and then utilizes google as the program of the computer. This technique then offers seniority to firm websites on the internet. It has been confirmed that snooping web-content for a small amount of time can obtain low ranks in terms of internet search, and this then becomes the foundation for this technique [28]. The approach of planning can be reduced to three main steps. The major step is capturing an image of the website in the user's application. The step that follows uses the optical character mechanisms for converting the image captured to text that is machine-readable. The third step entails inputting the text that is reborn into a research engine to obtain results and analyze the page's rank [20].

Pros: Overall, the GoldPhish does not lead to false positive. Also, it offers a zero-day snooping.

Cons: The GoldPhish technique slows down the process of rendering a webpage. It is also vulnerable to attacks on Google's PageRank algorithm and the search service [14].

4) **Character based anti-snooping approaches:** Many times when hackers attempt to steal data from users, they do so by enticing the users to click on URI and hyperlinks that they have embedded in snooped emails. A hyperlink is made up of the format: <ahref='URI'. Anchor text, n >.
The URI (Universal Resource Identifiers) offers the real link to where the user shall be guided. The anchor text refers to the text displayed in the web browser and stands for the visual connection [30]. This approach makes use of hyperlink characteristics in detecting the links that are snooped. LinkGuard refers to a tool that implements and executes this methodology. After many snooping websites are analyzed, the hyperlinks are then grouped into different categories. To detect the snooping websites, the LinkGuard tool initially obtains the DNS names, and if the terms are not the same, then it is a snooping attack [6].

This methodology's weakness is that it can lead to false positives because it uses decimal IP addresses that are dotted in place of domain names. Nonetheless, this may be appropriate in some special situations [12].

6 Conclusion and Future Work

With the advancement in technology, the recent years have come with a drastic increase in the sophistication and the number of email snooping attacks. Several techniques have been developed to detect and prevent snooping attacks such as attribute-based anti-snooping techniques, Identity-based anti-snooping techniques, content-based anti-snooping approaches, and character-based anti-snooping approaches. Visual similarity-based snooping involves sending large amounts of spoofed emails, asking the targeted users to click the links embedded in emails. By just a mere glance, the hyperlinks in the emails are generally challenging to suspect, and this makes it easy for the victim to click on them without their knowledge. Future work on the visual similarity snooping technique should entail creating improved ways to detect the malicious links and attachments in the snooped emails and deleting them. Another improvement that should be done on the method is applying machine learning techniques to make the visual similarity snooping technique adaptive.

References

[1] S. Afroz, R. Greenstadt, "PhishZoo: Detecting phishing websites by looking at them," *Detecting Phishing Websites By Looking at Them*, vol. 2, no. 6, pp. 1–8, 2015.

[2] G. AliMontazer, S. A. Yarmohammadi, "Detection of phishing attacks in Iranian e-banking using a fuzzy–rough hybrid system," *Applied Soft Computing*, vol. 35, pp. 482–492, 2015.

[3] F. Alkhatee, A. m. Manasrah, A. Al R. K Bsoul, "Bank web sites phishing detection and notification system based on semantic web technologies," *International Journal of Security & Its Applications*, vol. 6, no. 14, pp. 1–14, 2012.

[4] N. A. G. Arachchilage, S. Love, K. Beznosov, "Phishing threat avoidance behaviour: An empirical investigation," *Computers in Human Behavior*, vol. 60, no. 1, pp. 185–197, 2016.

[5] R. W. Ausen, W. J. Kopecky, "Extrusion die element, extrusion die and method for making multiple stripe extrudate," *U.S. Patent and Trademark Office*, vol. 9, no. 2, pp. 327–429, 2016.

[6] R. Butler, M. Butler, "Assessing the information quality of phishing–related content on financial institutions' websites," *Information & Computer Security*, vol. 26, no. 5, 2018.

[7] Y. L. Chi, I. C. Lin, H. C. Chen, "The features of phishing detection based on judgment user device," in *4th International Conference on Computer Technology and Science*, 2015.

[8] C. Crane, *20 Phishing Statistics to Keep You from Getting Hooked in 2019 - Hashed Out by The SSL Store*, Oct. 24, 2019. (`https://www.thesslstore.com/blog/20-phishing-statistics-to-keep-you-from-getting-hooked-in-2019/`)

[9] M. Deshmukh, S. K. Popat, "Different techniques for detection of phishing attack," *International Journal of Engineering Science and Computing*, vol. 7, no. 4, pp. 1–4, 2017.

[10] I. Drigă, C. Isac, "E-banking services-features, challenges and benefits," *Annals of the University of Petroani Economics*, vol. 14, pp. 49–58, 2014.

[11] R. Dhamija, J. D. Tygar, M. A. Hearst, "Why phishing works," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM, vol. 5, no. 3, pp. 581–590, 2006.

[12] M. Eoyang, "Beyond privacy and security: The Role of the telecommunications industry in electronic surveillance," *Security, Surveillance*, vol. 9, no. 3, p. 259, 2017.

[13] M. Hara, A. Yamada and Y. Miyake, "Visual similarity-based phishing detection without victim site information," vol. 1, no. 3, pp. 1–7, 2015.

[14] J. R. Hollenbeck, P. M. Wright, "Harking, sharking, and tharking: Making the case for post hoc analysis of scientific data," *Journal of Management*, vol. 5, no. 2, pp. 5–18, 2017.

[15] A. K. Jain, B. B. Gupta, "Phishing detection: analysis of visual similarity based approaches," *Security and Communication Networks*, pp. 1–21, 2017.

[16] B. Jongman, "Recent online resources for the analysis of terrorism and related subjects," *Perspectives on Terrorism*, vol. 13, no. 2, pp. 156–189, 2019.

[17] A. Kak, *Mounting Targeted Attacks for Cyber Espionage with Trojans and Social Engineering*, Purdue University, 2020.

[18] W. Kim, O. R. Jeong, C. Kim, J. So, "The dark side of the Internet: Attacks, costs and responses," *Information systems*, vol. 3, no. 36, pp. 675–705, 2011.

[19] C. C. Lee, C. H. Lin, M. S. Hwang, "Guessing attacks on strong–password authentication protocol," *International Journal of Network Security*, vol. 15, pp. 64–67, 2013.

[20] A.V. R. Mayuri, "Phishing detection based on visual–similarity," *International Journal of Scientific and Engineering Research*, vol. 3, no. 6, pp. 1–5, 2012.

[21] E. Medvet, E. Kirda and C. Krügel, "Visual-similarity-based phishing detection," July 6, 2018. (`http://www.eurecom.fr/en/publication/2515/detail/visual-similarity-based-phishing-detection`)

[22] S. Z. Nur, S. Deris , F. A. R. Mohd Faizal, F. Ahmad , I. S. W. D. Wan , K. Shahreen and S. Tole, "Phishing detection system using machine learning classifiers," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, pp. 1165–1171, 2020.

[23] P. Pasricha, S. Mehrotra, "Electronic crime in Indian banking," *Journal of Commerce and Management*, vol. 1, no. 11, pp. 7–14, 2014.

[24] J. M. Pavía, E. J. Veres-Ferrer, G. Foix-Escura, "Credit card incidents and control systems," *International Journal of Information Management*, vol. 32, no. 6, pp. 501–503, 2012.

[25] G. Sonowal, K. S. Kuppusamy, "PhiDMA - A phishing detection model with multi-filter approach," *Journal Of King Saud University-Computer and Information Sciences*, vol. 32, no. 1, pp. 99–112, 2020.

[26] I. W. Stats, "Internet world stats," *Internet World Stats*, Oct. 24, 2019. (`https://www.internetworldstats.com/stats.htm`)

[27] A. A. Tewari, "Recent survey of various defence mechanisms against phishing attacks," *Journal of Information Privacy and Security*, vol. 12, no. 2, pp. 3–13, 2016.

[28] C. Tian, M. L. Jensen, A. Durcikova, "Phishing susceptibility across industries: The differential impact of influence techniques," in *Proceedings of the 13th Pre-ICIS Workshop on Information Security and Privacy*, vol. 1, no. 1, pp. 1–20, 2018.

[29] G. Varshney, M. Misra, P. K. Atrey, "A survey and classification of web phishing detection," *Security and Communication Networks*, vol. 1, no. 9, pp. 6266–6284, 2016.

[30] A. Wright, S. Aaron, D. W. Bates, "The big phish: Cyberattacks against U.S. healthcare systems," *Journal of General Internal Medicine*, vol. 31, pp. 1115–1118, 2016.

[31] P. Yang, G. Zhao, P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, pp. 15196–15209, 2019.

[32] Y. Zhou, Y. Zhang, J. Xiao and Y. Wang, "Visual similarity based anti-phishing with the combination of local and global features," in *IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, vol. 1, no. 3, pp. 189–196, 2014.

Biography

George Mwangi Muhindi is a Master's Student at the Murang's University of Technology pursuing a Master's Degree in Information Technology. He holds a Bachelor of Business Information Technology from the Jomo Kenyatta University of Agriculture and Technology (JKUAT). His interests include Cyber Security, Machine Learning, and Software Development.

Dr. Geoffrey Mariga Wambugu is a Lecturer and Ag. CoD of Information Technology (IT) Department at the Murang's University of Technology. He obtained his BSc Degree in Mathematics and Computer Science from Jomo Kenyatta University of Agriculture and Technology in 2000, and his MSc Degree in Information Systems from the University of Nairobi in 2012. He holds a Doctor of Philosophy in Information Technology degree from JKUAT. His interests include Machine Learning and Text Analytics. Dr. Mariga has been involved in the design, development, and implementation of IT/ICT and Computer Science Curricula in different Universities and Colleges in Kenya.

Dr. Aaron Mogeni Oirere is a Lecturer and CoD of the Computer Science Department at Murang's University. He obtained his BSc. Degree in Computer Science from Periyar University in 2007, and his MSc. Degree in Computer Science from Bharathiar University in 2010. He holds a Ph.D. in Computer Science from Dr. Babasaheb Ambedkar Marathwada University. His interests include Database Management Systems, Hardware & Networking, Human-Computer Interface, Information Systems, Data Analytics, and Automatic Speech Recognition. He has presented papers in scientific conferences and has many publications in refereed journals.

Network Security Model for Multi-parallel Wireless Communication based on BMNS

Fengfei Kuang
(Corresponding author: Fengfei Kuang)
Minnan Science and Technology Institute
Nan An, Quanzhou, Fujian, China
(Email: 863898066@qq.com)
(Received July 7, 2019; revised and accepted Mar. 8, 2020)

Abstract

Communication networks play a critical role in industry and our daily lives. Network security is among the most concerned research topics in academic field, especially in the multi-parallel wireless communication for example 5G which is significant for the next communication channel. Theoretical models are limitedly reported from literature to examine the security in multi-parallel wireless communication network. This paper thus introduces a biology-based multidimensional network security (BMNS) model which uses hidden Markov chain model for investigating the security states. The parameters could be estimated and improved using Gibbs replacement method. The proposed model is able to describe the technical characteristics which could be used for the network security examination in the practical implementation.

Keywords: Biology-based Multidimensional Network Security (BMNS); Multi-parallel Network; Network Security; Security Model

1 Introduction

Communication networks play critical roles in our daily lives since our Internet and smart phone systems are mostly based on the wired or wireless communication channels [10, 16, 17]. 5G as a next communication media is attracting more and more attention. Next generation mobile networks alliance (NGMNA) has published a 5G white paper that covers the expectations, challenges and how the standards bodies, operators and vendors can accomplish to successfully enter the 5G decade [11]. 5G will embrace a high-speed environment from multiple access technologies, multi-layer networks, to large number of devices with billions of user interactions. Such enormous interactions will take advantage of 5G to municipalities such as energy and health from social organizations to public safety and defense [7]. 5G enables new services for all these users at low cost by providing a seamless and efficient communication and improve the way people interact with each other, with the final goal of improving people's lives [9]. Therefore, 5G network doesn't have the limitations to the radio access (RAN), but will encompass the whole network, including aspects as subscriber, policy and security management, core network and transport components [13].

Security is very critical in the 5G network as it is to move beyond delivering connectivity which uses security as a competitive advantage [3]. Therefore, different individual is able to seize the 5G opportunities in various purposes. A huge botnet formed by hacking into user devices in 4G could be used to mount large-scale DDOS attacks on websites, but in 5G world, that same botnet could be used to take out an entire network [6]. Network security is important as vast amounts of remote sensors and smart devices hooked up to global networks. For instance, it will radically increase the complexity of securing corporate networks from intruders and cyber criminals and the sheer amount of data being created by 5G networks will make it much more difficult to spot anomalies in user behavior resulting from hackers [1, 14].

In order to enhance the network security, this paper introduces a Biology-based Multidimensional Network Security (BMNS) model for a multi-parallel wireless communication network. BMNS conceptual model is a descriptive model that has abundant biological properties and technical features [5]. In this paper, based on BMNS conceptual model, a network security approach is established by using hidden Markov model (HMM) theory to quantify the safe state transition of BMNS through parameter estimation and model correction so as to achieve adaptive and robust for improving the network security.

The rest of this paper is organized as follows. Section 2 introduces the parameter estimation for BMNS. Three sub-sections are included in this section to illustrate the initial model mechanism, solution algorithm, and the estimations of parameters in the multi-parallel wireless communication. Section 3 reports on the improved mathematic model which uses Gibbs replacement method, proposed state stay approach, and correction of model with

theoretical analysis. Section 4 concludes this paper by giving the contributions and future research directions.

2 Parameter Estimation

A reasonable setting of the states sequence, the relationship between state transition, and observations are based on parameter estimation for BMNS. State probability distribution can be obtained when the system is steady through transition relationship between states and transition probabilities, resulting in randomness of state transition process. Markov transition theory can quantitatively analyze the limiting steady-state characteristics of this process by converting the continuously changing process into states that are linked by transition relationships of transition probabilities [15, 20]. Markov transition theory treats network operation process of BMNS as a series of specific states. According to the progress of safe-state time stages (precaution, detection and response, tolerance, and recovery), original data and initial model for BMNS parameter estimation will be provided.

2.1 Initial Model

Different initial models may generate different training results as the algorithm can obtain the model parameters when $P(O/\lambda)$ is the local maximum. It's meaningful to select a good initial model for the final local maximum that is close to the global maximum [12]. Generally, the initialization of π and A has a little impact on the results, so their values can be initialized randomly or uniformly as long as the constraints $0 \le a_{ij} \le 1$, $\sum_j^N a_{ij} = 1$, $0 \le \pi_i \le 1$ and $\sum_i \pi_i = 1$ are satisfied. However, the initialization of B has a profound impact on the trained HMM, initialize approach for the value is adopted.

The initial model of BMNS mathematical model can be established according to functional features of the conceptual model. Different functions of safe mechanisms contain precaution, detection and response, tolerance, and recovery, and transition relationship between states can refer to [4]. According to the completion progress of different time stage states, there are a continuous process of each state includes three phases, namely initializing, processing and completing. Thus, based on HMM theory and BMNS state transition relationship, there is an initial model $\lambda = (N, M, \Pi, A, B)$:

1) $N = 4$ means four states, namely precaution state, detection and response state, tolerance state and recovery state, and let S_1, S_2, S_3, S_4 denote these states respectively;

2) $M = 3$ means three observations, namely Initializing (I), Processing (P) and Completing (C), where the observation at time t is $O_t \in \{\mathrm{I}, \mathrm{P}, \mathrm{C}\}$;

3) The model starts from S_1, and ends at S_4. Thus, $\Pi = (\pi_1, \pi_2, \pi_3, \pi_4) = (1, 0, 0, 0)$;

4) According to the state transition relationship of the model, we set

$$A = (a_{ij})_{4\times 4} = \begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \end{pmatrix};$$

5) $B = (b_{jk})_{4\times 3} = \begin{pmatrix} 0.1 & 0.5 & 0.4 \\ 0.6 & 0.1 & 0.3 \\ 0.2 & 0.4 & 0.4 \\ 0.2 & 0.1 & 0.7 \end{pmatrix}$

Where b_{jk} represents the probability $(1 \le j \le N, 1 \le k \le M)$ when the observation of the model in state is. In parameter estimation process, the probability S_j with observation O_k when the state is converted from S_i to S_j is introduced. Thus, we can extend B shown as Table 1.

It is assumed that the operation states of network are always in completing phase for BMNS. Thus, we choose $O = (O_1, O_2, O_3) = (C, C, C)$ as samples and use Baum-Welch algorithm to estimate parameters for BMNS mathematical model [19]. Firstly, N, M, Π, A and B of initial model serve as input to estimate $\bar{\lambda}$ that is composed of $\bar{\pi}$, $\bar{a}_{ij}$ and $\bar{b}_{ij}$. Then $\bar{\pi}$, $\bar{a}_{ij}$ and $\bar{b}_{ij}$, as new inputs, are used to re-estimate the parameters. This process is repeated until a $\lambda = (\Pi, A, B)$ is obtained to maximize $P(O/\lambda)$.

2.2 Solution Algorithm

Baum-Welch algorithm is used for solving HMM parameter estimation problem [18]. Given the sequence of observations $O = O_1, O_2, \cdots, O_T$, $\lambda = (\Pi, A, B)$ can be determined to maximize $P(O/\lambda)$. Re-estimation formula of Baum-Welch algorithm is shown as follows.

$$\bar{\pi} = \xi_1(i) \tag{1}$$

$$\bar{a}_{ij} = \sum_{t=1}^{T-1} \xi_t(i,j) \Big/ \sum_{t=1}^{T-1} \xi_t(i) \tag{2}$$

$$\bar{b}_{jk} = \sum_{t=1, O_t=V_k}^{T} \xi_t(j) \Big/ \sum_{t=1}^{T} \xi_t(j) \tag{3}$$

Where

$$\begin{aligned} \xi_t(i) &= P(O, q_t = S_i/\lambda) \\ &= \sum_{j=1}^{N} \xi_t(i,j) = \alpha_t(i)\beta_t(i)/P(O/\lambda) \end{aligned}$$

is the probability that Markov chain is in state S_i at time t; $\xi_t(i,j) = P(O, q_t = S_i, q_{t+1} = S_j/\lambda) = [\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)]/P(O/\lambda)$ is the probability that Markov chain is in state S_i at time t and next in state S_j at time $t+1$; $\alpha_t(i) = P(O_1, O_2, \cdots, O_t, q_t = S_i/\lambda), 1 \le t \le T$ is the forward variable, and its recursive process is shown as follows:

Table 1: The extended B

i	$b_{ij}(O_k)$											
	$j=1$			$j=2$			$j=3$			$j=4$		
	$k=1$	$k=2$	$k=3$	$k=1$	$k=2$	$k=3$	$k=1$	$k=2$	$k=3$	$k=1$	$k=2$	$k=3$
1	0.1	0.5	0.4	1/3	1/3	1/3	1/3	1/3	1/3	0	0	0
2	0	0	0	0.6	0.1	0.3	1/3	1/3	1/3	1/3	1/3	1/3
3	0	0	0	0	0	0	0.2	0.4	0.4	1/3	1/3	1/3
4	0	0	0	0	0	0	0	0	0	0.2	0.1	0.7

1) Initialization $\beta_T(N)=1, \beta_T(j)=0\,(j\neq N)$.

2) Recursion formula $\beta_t(i)=\sum_{j=1}^{N}\beta_{t-1}(j)a_{ij}b_{ij}(O_{t-1})$ $(t=T-1,T-2,\cdots,1; i,j=1,2,\cdots,N)$.

3) The final result $P(O/\lambda)=\sum_{i=1}^{N}\beta_1(i)$ is the backward variable, and its recursive process is shown as follows:

   a. Initialization $\beta_T(N)=1, \beta_T(j)=0\,(j\neq N)$.

   b. Recursion formula

$$\beta_t(i)=\sum_{j=1}^{N}\beta_{t-1}(j)a_{ij}b_{ij}(O_{t-1})$$
$$(t=T-1,T-2,\cdots,1; i,j=1,2,\cdots,N)$$

   c. The final result $P(O/\lambda)=\sum_{i=1}^{N}\beta_1(i)$.

In practical applications, if the functions (or states) of the model are added or changed, this algorithm can be generalized according to HMM module training methods. For a training dataset, such as D_A, D_B and D_C, λ_A, λ_B and λ_C are generated by Baum-Welch algorithm. The number of relevant transition, the number of vectors and the number of states to flexibly reflect each model's parameters of D_A+D_B, D_A+D_C, D_B+D_C and $D_A+D_B+D_C$, as long as corresponding numerators and denominators are added respectively. Therefore, the process of HMM parameter estimation could be adaptive.

2.3 Parameter Estimations

This paper uses VC++ to realize Baum-Welch algorithm, where $\bar{\pi}$, $\bar{a}_{ij}$, $\bar{b}_{ij}$, and can be calculated by Equations (1), (2), (3) and (4).

$$P(O/\lambda)=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)\,a_{ij}b_{ij}(O_{t+1})\,\beta_{t+1}(j),$$
$$1\leq t\leq T-2 \qquad (4)$$

As shown in Figure 1, $P(O/\lambda)$ gradually converges with the number of cycles increasing.

The final results of Π, A, B and $P(O/\lambda)$ are shown in Table 2.

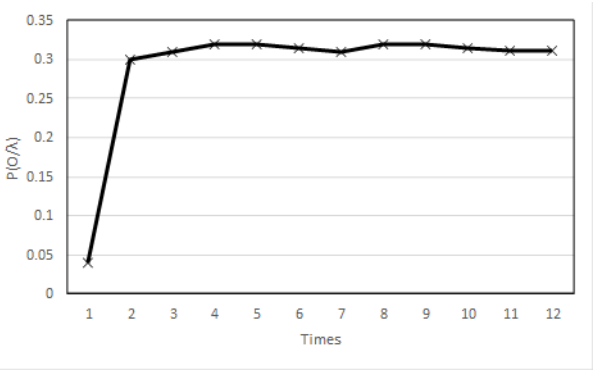


Figure 1: Relationship between $P(O/\lambda)$ and repeating times

During the running process, data related to parameter estimation are shown in Table 3.

From Table 3, it could be observed that with the increase of iteration time, the deviation of $P(O/\lambda)$ is not significant. That shows the reliability of network which attributes the Baum-Welch algorithm that figures out the $P(O/\lambda)$ considering λ. For the medium variables, it is found that $\alpha_t(i)$ and $\beta_t(i)$ are decreasing when the iteration times increase. However, the values remain stable when $n\to\infty$.

3 Improved Multidimensional Mathematical Model

Based on Baum-Welch algorithm, BMNS mathematical model can exactly reflect the transition relationship of the conceptual model. However, camped probability of effective states of Markov chain represented by π and A stay in different time stages cannot be clearly presented. Therefore, this section focuses on of the improvement of BMNS model. Two approaches can be adopted:

1) Using Gibbs distribution to replace HMM of Markov chain;

2) Increasing state duration.

Table 2: Parameter estimation results

$P(O/\lambda)$	Π	A				B		
0.33333	$(1,0,0,0)$	4.46357e-007	0.327284	0.672715	0	0	0	1
		0	1.93327e-007	4.22823e-007	0.999999	0	0	1
		0	0	3.249e-007	1	0	0	1
		0	0	0	1	0	0	1

Table 3: Result variables

Cycles	$P(O/\lambda)$	A				B		
1	0.0348	0.104	0.311	0.584	0	0	0	1
		0.088	0.867	0	0	1	0	1
		0.076	0.923	0	0	1	0	1
		0	1	0	0	1	0	1
2	0.299	0.029	0.323	0.647	0	0	0	1
		0.026	0.961	0	0	1	0	1
		0.02	0.979	0	0	1	0	1
		0	1	0	0	1	0	1
3	0.324	0.009	0.326	0.664	0	0	0	1
		0.008	0.987	0	0	1	0	1
		0.006	0.993	0	0	1	0	1
		0	1	0	0	1	0	1
4	0.33	0.003	0.326	0.67	0	0	0	1
		0.002	0.995	0	0	1	0	1
		0.002	0.997	0	0	1	0	1
		0	1	0	0	1	0	1
n	0.333	0.0003	0.327	0.672	0	0	0	1
		0.0003	0.999	0	0	1	0	1
		0.0002	0.999	0	0	1	0	1
		0	1	0	0	1	0	0

3.1 Gibbs Replacement Method

Gibbs distribution is used to describe the sequence of states of HMM [2, 8]. It can replace Markov chain represented by π and A so as to obtain a complete HMM. In HMM, by adopting Gibbs distribution, the probability generated by the sequence of states $S = q_1, q_2, \cdots, q_T$ is:

$$P(S/\lambda) = \exp[-U(S)]/Z. \quad (5)$$

$$Z = \sum_{\cup S} \exp[-U(S)]. \quad (6)$$

where $U(S)$ is the energy function, and Z is a normalized term, namely

Therefore, Gibbs distribution is determined by its energy function $U(S)$. For one-dimensional first-order Markov random field, the general form of the energy function $U(S)$ is:

$$U(S) = \sum_{t=1}^{T} h(q_t) + \sum_{t=2}^{T} g(q_{t-1}, q_t). \quad (7)$$

where h and g are any real functions. If Gibbs distribution is used to represent Markov chain, there is:

$$U(S) = \sum_{j=1}^{N} \hat{\pi}_j J_j(q_1) + \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \hat{a}_{ij} J_{ij}(q_{t-1}, q_t). \quad (8)$$

$$J_j(q_1) = \begin{cases} 1, \text{if} q_1 = \theta_j \\ 0, \text{otherwise} \end{cases} \quad (9)$$

$$J_{ij}(q_{t-1}, q_1) = \begin{cases} 1, \text{if} q_{t-1} = \theta_i, q_t = \theta_j \\ 0, \text{otherwise} \end{cases} \quad (10)$$

$$\sum_{j=1}^{N} e^{-\hat{\pi}_j} = 1 \quad (11)$$

$$\sum_{j=1}^{N} e^{-\hat{a}_{ij}} = 1, i = 1, \cdots, N \quad (12)$$

Gibbs distribution can be described by parameters $\hat{\pi}_j$ and $\hat{a}_{ij}$. Clearly, the relationships between these parameters and π as well as A are

$$\hat{\pi}_j = -\lg \pi_j, j = 1, \cdots, N \quad (13)$$

$$\hat{a}_{ij} = -\lg a_{ij}, i, j = 1, \cdots, N \quad (14)$$

Therefore, if the sequence of states of HMM that is described by Gibbs distribution presented by energy function $U(S)$ in equation Equation (8) is totally equal to the sequence of states of Markov chain that is described by π and A. However, given the general formula of $U(S)$ as shown in equation Equation (7), there is a wide variety of Gibbs distribution to describe the sequence of states. For example, energy function can be selected under one-dimensional nearest field situation:

$$U(S) = \sum_{j=1}^{N} \sum_{t=1}^{T} \hat{a}_j J_j(q_t) + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T} \hat{\beta}_{j-1} J_j(q_{t-1}, q_t)$$

where $\hat{a}_j$ is an external field parameter, $\hat{\beta}_{j-1}$ is a conjoint intensity function, and they are used together to describe energy function $U(S)$ or related Gibbs distribution. Therefore, $\left(\hat{\alpha}, \hat{\beta}, B\right)$ is a set of parameters of HMM by introducing Gibbs distribution, where $\hat{\alpha}$ and $\hat{\beta}$ are used to describe Gibbs distribution. The correction process is based on Baum-Welch algorithm, forward function and backward function are defined to effectively solve the calculation issue of $P(O/\lambda)$:

$$P(O/\lambda) = \sum_{\cup S} P(O/S, \lambda) P(S, \lambda). \quad (15)$$

In Equation (5) of $P(S/\lambda)$, $P(O/\lambda)$ cannot be obtained by direct calculation due to the computational complexity. However, this correction process is not suitable for BMNS model.

3.2 State Stay

In BMNS model, the probability that d observations are generated gradually in state S_i

$$p_i(d) = (a_{ij})^d (1 - a_{ij}). \tag{16}$$

The probability $p_i(d)$ describes state duration of state S_i. It is an exponential distribution, and its maximum value is located in $d = 0$. However, this characteristic does not accord with state transition process of BMNS. Therefore, the basic idea for correction of BMNS model is to use the non-exponential distribution $P_i(d)$ to describe state duration. That means parameter sets (Π, A and B) are used to describe Markov chain are corrected by introducing the probability $P_i(d)$ that is used to describe state duration using the follow methods:

3.2.1 Non-Parameter Estimation Method

Non-parameter estimation is widely used. In Markov chains, let $a_{ij} = 0$, and the probability distribution $P_i(d)$ of state duration is introduced, where $d = 1, \cdots, D$ and D is the longest duration of all the possible state stay. Thus, the process of inputting observation sequences produced by HMM follows that initial state q_i is selected according to π_i, and state duration d_1 that produces d_1 observations $O_1, O_2, \cdots, O_{d1}$ is determined according to $P_{q1}(d)$. The probability is $\prod_{t=1}^{d_1} b_{q_1}(O_t)$. According to $a_{q_1 q_2}$, state q_2 is selected. The above process is repeated until all the sequence of observations $O = O_1, O_2, \cdots, O_T$ is generated. To calculate $P(O/\lambda)$, forward variables are defined as $\alpha_t(i) = P(O_1, O_2, , O_t, \text{state} S_i/\lambda \text{endsattimet})$. Therefore, $P(O/\lambda) = \sum_{i=1}^{N} a_T(i)$ is similar to classical HMM. To train this corrected HMM, we have to redefine three forward and a backward variables, and derive re-estimation formula to estimate parameters. Besides, although the performance is better compared with classical HMM, parameter $P_i(d)$ of HMM is added. Especially, in order to estimate reliable parameter $P_i(d), i = 1, \cdots, N, d = 1, \cdots, D$, there is a request for more training data.

3.2.2 Parameter Estimation Method

In order to estimate $P_i(d)$ with limited training data, parameter estimation method is proposed to describe state duration. The specific value of $P_i(d)$ cannot be estimated directly, but $P_i(d)$ is assumed to be drawn from a certain distribution and $P_i(d)$ can be estimated through parameters used to describe the distribution. Assume $P_i(d)$ is drawn from the Gamma distribution:

$$P_i(d) = \frac{\eta_i^{V_i} d^{V_i - 1} e^{-\eta_i^d}}{\Gamma(V_i)} \tag{17}$$

Thus, the estimation of $P_i(d)$ is converted to estimate parameters v_i and η_i of the Gamma distribution. For the revised HMM, parameter set used to describe Markov chain is (Π, A, V, η), where $V = (v_1, \cdots, v_N)$ and $\eta = (\eta_1, \cdots, \eta_N)$. According to the forward-backward algorithm and Baum-Welch algorithm, it is easy to derive calculation formula of probability $P(O/\lambda)$ and re-estimation formula of other parameters.

When this method is used to obtain $P_i(d)$, although we can avoid the strict requirement for the volume of training data used to estimate value of $P_i(d)$ by non-parameter method, more computational power is needed. And this method is not suitable for all states, because $P_i(d)$ is assumed to be drawn from a certain distribution manually.

3.2.3 Upper Limit and Lower Limit Estimation Method

Except parameter estimation method and non-parameter estimation method, upper limit and lower limit estimation method is a satisfying method to correct classical HMM by only estimating upper parameters u_i and lower parameters l_i of state duration of each state S_i. The parameter set of this revised HMM is (Π, A, L, U, B), where Π, A and B are parameters of classical HMM while $L = (l_1, \cdots, l_N)$ and $U = (u_1, \cdots, u_N)$ are new parameters that describe minimum and maximum (lower limit and upper limit) of each state duration. Let $d(i)$ denote the length of state duration of S_i. Thus, for a state sequence $S = q_1, q_2, \cdots, q_T$, if assume Markov chain starts from state S_l and ends at state S_T, then:$S = \underbrace{\theta_1 \cdots \theta_1}_{d(1)} \underbrace{\theta_2 \cdots \theta_2}_{d(2)} \cdots \underbrace{\theta_N \cdots \theta_N}_{d(N)}$. Estimation method of newly added parameter L and U is following: Let d_{ik} denotes state duration of S_i in optimal state sequence Q_k^* calculated by Viterbi algorithm of $k - th$ training sequences $O^{(k)} = {O_1}^{(k)}, {O_2}^{(k)}, \cdots, {O_{Tk}}^{(k)}$, and assume there are K training sequences, so:

$$\begin{aligned} l_i &= \min_{k=1}^{K} \{\max[d_{ik}/T_k, 1/T_k]\}, i = 1, 2, \cdots, N \\ u_i &= \max_{k=1}^{K} \{\max[d_{ik}/T_k, 1/T_k]\}, i = 1, 2, \cdots, N \end{aligned}$$

3.3 Correction of Model

We adopt upper limit and lower limit estimation method to correct HMM of BMNS. Based on the model, we set $k = 3$, $O^{(1)} = (\mathrm{C}, \mathrm{C}, \mathrm{C})$, $O^{(2)} = (\mathrm{C}, \mathrm{C}, \mathrm{C}, \mathrm{C})$, $O^{(3)} = (\mathrm{C}, \mathrm{C}, \mathrm{C}, \mathrm{C}, \mathrm{C})$, Q_k^*

1) Initialization:

$$\begin{aligned} \delta_1(i) &= \pi_i b_{ij}(O_t), 1 \le i \le N \\ \varphi_1(i) &= 0, 1 \le i \le N \end{aligned}$$

2) Recursion:

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_{ij}(O_t), 2 \le t \le T, \quad 1 \le j \le N$$

$$\varphi_t(j) = \arg\max_{1\le i\le N}[\delta_{t-1}(i)\, a_{ij}], 2 \le t \le T, \quad 1 \le j \le N$$

3) End:

$$P^* = \max_{1\le i\le N}[\delta_T(i)]$$
$$q_T^* = \arg\max_{1\le i\le N}[\delta_T(i)]$$

4) Optimal sequence of states:

$$q_t^* = \varphi_{t+1}\left(q_{t+1}^*\right), t = T-1, T-2, \cdots, 1$$

The obtained three optimal state sequence corresponding to three observation sequence are: $Q_1^* = (S_1, S_3, S_3)$, $Q_2^* = (S_1, S_3, S_4, S_4)$ and $Q_3^* = (S_1, S_3, S_4, S_4, S_4)$.

And based on Equation (6) and Equation (7), $L = (l_1, l_2, l_3, l_4) = (1/5, 1/5, 1/5, 1/3)$ and $U = (u_1, u_2, u_3, u_4) = (1/3, 1/3, 1/3, 3/5)$ can be obtained. Thus, the whole BMNS mathematical model $\lambda_{B-MNS} = (\Pi, A, L, U, B)$ can be obtained, where

1) $\Pi = (1, 0, 0, 0)$

2) $A = \begin{pmatrix} 4.46357^{-7} & 0.327284 & 0.762715 & 0 \\ 0 & 1.93327e^{-7} & 4.22823e^{-7} & 0.999999 \\ 0 & 0 & 3.249e^{-7} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

3) $L = (l_1, l_2, l_3, l_4) = (1/5, 1/5, 1/5, 1/3)$

4) $U = (u_1, u_2, u_3, u_4) = (1/3, 1/3, 1/3, 3/5)$

5) $B = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

4 Conclusion

Based on Markov transition theory, the network operation process of BMNS is converted into a serial of specific states. States make a transition according to the completion progress of safe-state time stages (precaution, detection and response, tolerance, and recovery). This paper uses HMM theory to quantify the process of BMNS state transition and uses Baum-Welch algorithm to estimate its parameters. Upper limit and lower limit estimation methods are then adopted to increase state duration and to establish the reasonable BMNS model.

Future research will be conducted as follows. Firstly, comparison analysis or studies will be carried out to examine the performance with other approaches such as stochastic models and chaotic models. Secondly, the model testing and results analysis will be focused after introducing the theoretical models. Some key performance indicators will be examined like mean response rate, Hit-velocity in BMNS, and HTTP response rate in the multi-parallel wireless communication networks for example 5G.

Acknowledgement

Authors would like to acknowledge the key educational revolution project (No. MKJG-2018-001) entitled "Investigation of Teaching Mode for Network Talent in the context of Applied Colleges" from Minnan Science and Technology Institute.

References

[1] B. S. Alghamdi, M. Elnamaky, M. A. Arafah, M. Alsabaan, and S. H. Bakry, "A context establishment framework for cloud computing information security risk management based on the stope view," *International Journal Network Security*, vol. 21, no. 1, pp. 166–176, 2019.

[2] T. Austin *et al.*, "The structure of low-complexity gibbs measures on product spaces," *The Annals of Probability*, vol. 47, no. 6, pp. 4002–4023, 2019.

[3] M. Behi, M. GhasemiGol, and H. V. Nejad, "A new approach to quantify network security by ranking of security metrics and considering their relationships," *International Journal Network Security*, vol. 20, no. 1, pp. 141–148, 2018.

[4] C. M. Chen, D. J. Guan, Y. Z. Huang, and Y. H. Ou, "Anomaly network intrusion detection using hidden markov model," *International Journal of Innovative Computing, Information and Control*, vol. 12, pp. 569–580, 2016.

[5] M. J. Faghihniya, S. M. Hosseini, and M. Tahmasebi, "Security upgrade against rreq flooding attack by using balance index on vehicular ad hoc network," *Wireless Networks*, vol. 23, no. 6, pp. 1863–1874, 2017.

[6] S. Godi and R. Kurra, "Novel security issues and mitigation measures in cloud computing: An indian perspective," *International Journal of Computer Applications in Technology*, vol. 58, no. 4, pp. 267–287, 2018.

[7] A. S. Khan, J. Abdullah, N. Khan, A. A. Julahi, and S. Tarmizi, "Quantum-elliptic curve cryptography for multihop communication in 5G networks," *International Journal of Computer Science and Network Security (IJCSNS'17)*, vol. 17, no. 5, pp. 357–365, 2017.

[8] A. Levy, H. C. Gibbs, and D. Boneh, "Stickler: Defending against malicious content distribution networks in an unmodified browser," *IEEE Security & Privacy*, vol. 14, no. 2, pp. 22–28, 2016.

[9] S. Li, L. D. Xu, and S. Zhao, "5g internet of things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[10] I. C. Lin and T. C. Liao, "A survey of blockchain security issues and challenges," *International Journal Network Security*, vol. 19, no. 5, pp. 653–659, 2017.

[11] NGMN Alliance, "Next generation mobile networks," *White Paper*, vol. 163, 2015.

[12] B. Rashidi, C. Fung, and E. Bertino, "Android resource usage risk assessment using hidden markov model and online learning," *Computers & Security*, vol. 65, pp. 90–107, 2017.

[13] S. Sekander, H. Tabassum, and E. Hossain, "Multi-tier drone architecture for 5G/B5G cellular networks: Challenges, trends, and prospects," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 96–103, 2018.

[14] U. K. Singh and C. Joshi, "Information security risk management framework for university computing environment," *International Journal Network Security*, vol. 19, no. 5, pp. 742–751, 2017.

[15] S. Sun, Y. Li, W. S. T. Rowe, X. Wang, A. Kealy, and B. Moran, "Practical evaluation of a crowdsourcing indoor localization system using hidden markov models," *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9332–9340, 2019.

[16] Y. Sun, M. Brazil, D. Thomas, and S. Halgamuge, "The fast heuristic algorithms and post-processing techniques to design large and low-cost communication networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 375–388, 2019.

[17] S. M. R. Taha and Z. K. Taha, "Eeg signals classification based on autoregressive and inherently quantum recurrent neural network," *International Journal of Computer Applications in Technology*, vol. 58, no. 4, pp. 340–351, 2018.

[18] C. O. Tinubu, D. O. Aborisade, A. S. Sodiya, S. A. Onashoga, and M. A. Ganiyu, "Towards detecting credit card frauds using hidden markov model," *Journal of Computer Science and Its Application*, vol. 26, no. 2, pp. 54–63, 2019.

[19] C. Wang, K. Li, and X. He, "Network risk assessment based on baum welch algorithm and HMM," *Mobile Networks and Applications*, 2020. DOI: 10.1007/s11036-019-01500-7.

[20] Z. G. Wu, P. Shi, Z. Shu, H. Su, and R. Lu, "Passivity-based asynchronous control for markov jump systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 2020–2025, 2016.

Biography

Mr. Fengfei Kuang is an Associate Professor in Minnan Science and Technology Institute, Fujian, China. His research covers Network Information Security, Computer Network Application Technologies, and Big Data. He has published several papers in international journals and conferences. He participated several key project from national and provincial departments in recent years.

Malicious Attack Detection Algorithm of Internet of Vehicles based on CW-KNN

Peng-Shou Xie, Cheng Fu, Tao Feng, Yan Yan, and Liang-Lu Li
(Corresponding author: Cheng Fu)

School of Computer and Communications, Lanzhou University of Technology
287 Lan-gong-ping Road, Lanzhou, Gansu 730050, China
(Email: 452708186@qq. com)
(Received Nov. 29, 2019; revised and accepted Mar. 21, 2020)

Abstract

With the wide application of multiple wireless communication technologies, vehicle nodes realize the connection of various networks such as WiFi, Bluetooth, 802.11p, LTE-V2X, and 5G. The attacker accesses the car's internal network through wireless communication,install malware for malicious attacks, these malicious attacks interfere with normal vehicle communication, spoofing or tapmer information, which will seriously threaten the security of the Internet of Vehicles. Therefore, this paper studies the main threats of malicious attacks on the Internet of Vehicles, extracts their malicious attack features, weights these features in combination, and proposed CW-KNN, which is a malicious attack detection algorithm suitable for Internet of Vehicles. Simulation experiments prove the effectiveness of the proposed algorithm.

Keywords: Combined Weight; CW-KNN; Internet of Vehicles; Malicious Attack Detection; Malware

1 Introduction

In the United States, the research work on the Internet of Vehicles (IoV) is based on Wireless Access in Vehicular Environment (WAVE) in Dedicated Short Range Communications(DSRC). The use of WAVE requires the construction of a dedicated service base station of IoV, this has greatly limited the popularity of IoV. But in China, in the 5G environment, vehicle nodes in the IoV rely on cellular wireless communication technology to communicate, and the related information is presented to the user through the upper-layer application. Huawei has established an LTE-V network and developed a communication chip. By loading a SIM card into a car, real-time communication services between cars can be achieved.

IoV is a part of wireless communication, wireless communication is generally integrated in vehicle systems, the CW-KNN detection algorithm proposed in this paper can also be integrated to protect the safe of IoV. Attackers installing malware can cause significant threats to IoV. The malicious attacks in this paper are active attacks, and the main threats are the following three aspects.

Denial of Service (DoS): Malware can interfere or block communication, causing vehicle nodes to fail to establish communication within the receiving range;

Spoofing: IoV's application technology requires accurate and timely access to application data. The attacker faked the relevant information and sent it, causing the vehicle to receive the wrong information, causing the driver to make abnormal behaviors, posing a certain threat to driving.

Tapmer: Malware can tamper information, each vehicle in IoV can be used as a terminal or relay node, information sent or received by them may be tampered, this will bring more scams and cause huge losses to the user.

It turns out that tapmer is easier than spoofing. Overall,malware will affect the normal function of the system, seriously affect driving safety, and even cause traffic accidents.

In terms of security of IoV, [15] proposed data falsification attack detection using hashes for enhancing network security and performance by adapting contention window size to forward accurate information to the neighboring vehicles in a timely manner. [20] in order to analyze the virus propagation under the road environment mixed with Cooperative Adaptive Cruise Control (CACC) vehicles andcommon vehicles, considering the interaction among traffic flow, information flow and virus propagation, CACC vehicle virus infection probability is calculated and the dynamic model of virus propagation is built. [22] aimed at the problem of security under the internet of vehicles environment, combining K area with fake names anonymous technology, a kind of improved Privacy Preservation Algorithm-Internet of Vehicles (PPA-IOV) privacy protection algorithm is formed. at the same time, researchers have also conducted related research on protocol and model strategies [8, 24, 25]. In

terms of malicious attack detection, [1] proposed a solution to the problem of detecting semantic attacks in data based on hybrid automata implementation state constraints. [12] proposed a network intrusion detection model based on K-nearest neighbor(KNN)algorithm of extreme learning machine Extreme Learning Machine (ELM)feature mapping. [9] proposed a semi-supervised fuzzy kernel clustering algorithm based on quantum artificial fish group.

Although researchers have recently proposed many detection methods [2–7, 11, 13, 14, 17–19], these detection methods are not very suitable for the IoV. In the above, we have proposed the main threats of the IoV, which have corresponding attack features. Traditional malicious attack detection methods treat the feature contributions of the samples as the same, and do not weight the features from these threats. The direct use in the IoV will reduce the detection accuracy.

The main technical contributions of this paper are as follows. First, a specific method for establishing a simulated attack dataset of IoV is proposed, which can provide support for further research on the detection technology of the malicious attack of IoV. Second, the Combination Weight-KNN (CW-KNN) detection algorithm is proposed, which makes up for the lack of a malicious attack detection method in IoV.

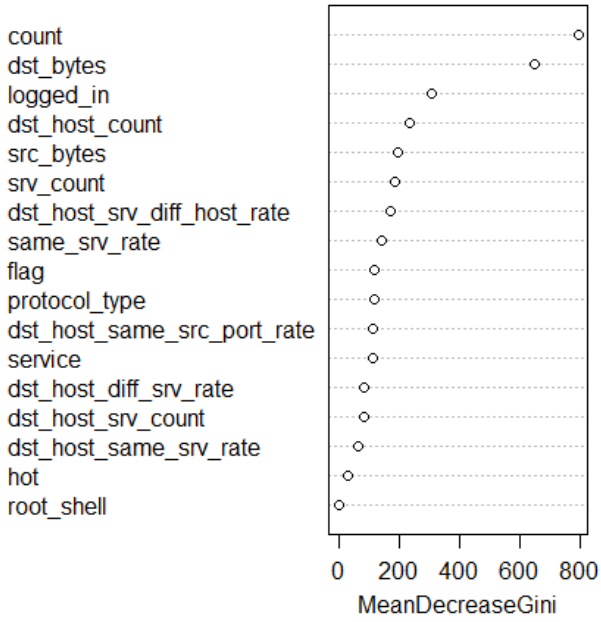


Figure 1: Feature contribution

2 Building a Simulated Attack Dataset of IoV

2.1 Feature Selection

The KDD CUP 99 [16]dataset marks each network connection as normal or abnormal. These anomaly types are further subdivided into 4 categories and a total of 39 attack types. A total of 22 attack types appeared in the training set, while the remaining 17 appeared only in the testing set. The criterion for evaluating intrusion detection is the ability to detect unknown attack types. KDD CUP 99 can well test the generalization power and applicability of the classification algorithm. It is also a recognized standard data set in the field of anomaly intrusion detection.

As the real-world malicious attack data set of IoV cannot be obtained, we improved KDD CUP 99 to obtain the simulation data set for experiments. The specific process is as follows.

The first step is to prune the original data set. There are 41 features in original KDD CUP 99 dataset. If all 41 features are used, this will lead to inaccurate and time-consuming results. Therefore, it is necessary to specifically remove some redundant features or low-important features. For example, "num_outbound_cmds" and "is_hot_login", The values are the same and they are all 0, So delete them.

The second step is to obtain the corresponding features of the malicious attack of the IoV. We studied the main threats to the IoV, and got the corresponding features. Some of these features are shown in Table 1.

Table 1: Feature contribution

The main malicious attacks on IoV	**Features**
Malicious code implantation	protocol_type, service, src_bytes, srv_count, count *etc.*
Spoofing	hot, root_shell, logged_in, num_access_files, flag *etc.*
Tamper	is_hot_login, is_guest_login, num_failed_logins *etc.*
Denial of service	src_bytes, dst_host_count, dst_host_srv_count *etc.*
Signal playback	dst_host_same_srv_rate, dst_host_same_src_port_rate *etc.*

The third step is to further optimize the selection of features. In order to avoid feature selection being too subjective in the previous section, and to make the selection persuasive, the Random Forest was used to evaluate the feature importance. Random forest can find out the degree of contribution of each feature to each tree, then take the average value, and finally compare the degree of contribution between features. the degree of contribution is usually measured using the Gini index as an evaluation indicator. as shown in Figure 1.

Finally, after many experiments, we selected 17 features, as shown in Table 2. We use the data set created by these 17 features as the simulation dataset for experiments.

Table 2: Final selected feature

Number	Feature name	Description	Types
1	protocol_type	Network protocol type	Discrete
2	service	The network service type of the target's host	Discrete
3	flag	Connected to a normal or incorrect state	Discrete
4	src_bytes	The number of bytes of data from source host to target host	Continuous
5	dst_bytes	The number of bytes of data from target host to source host	Continuous
6	hot	Number of times to access system sensitive files and directories	Continuous
7	logged_in	Successful login or not	Discrete
8	root_shell	Get superuser privileges or not	Discrete
9	count	The number of connections to the same target host as the current connection in the last two seconds	Continuous
10	srv_count	The number of connections with the same service as the current connection in the past two seconds	Continuous
11	same_srv_rat	Percentage of connections with the same service as the current connection in the last two seconds of a connection with the same target host	Continuous
12	dst_host_count	Of the top 100 connections,the number of connections with the same target host as the current connection	Continuous
13	dst_host_srv _count	Of the top 100 connections,the number of connections with the same target host and the same service as the current connection	Continuous
14	dst_host_same _srv_rate	Of the top 100 connections, percentage of connections with the same target host and the same service as the current connection	Continuous
15	dst_host_diff _srv_rate	Of the top 100 connections, percentage of connections with the same target host as the current connection but different services	Continuous
16	dst_host_same _src_port_rate	Of the top 100 connections, the percentage of connections with the same target host and the same source port as the current connection	Continuous
17	dst_host_srv _diff_host_rate	Of the top 100 connections, the current connection has the same target host and the same service. the percentage of connections with different source hosts from the current connection	Continuous

2.2 Data Preprocessing

To make the experiment more accurate, the data needs to be pre-processed before the experiment.

Numeric: One-hot encoding for the some features. for example,encoding "tcp", "udp", "icmp" as "0", "1", "2".

Standardization: Sij is the value normalized by the Xij value, as shown in Equations (1), (2), and (3).

$$S_{ij} = \frac{X_{ij} - AVG_j}{STAD_j} \tag{1}$$

$$AVG_j = \frac{X_{1j} + X_{2j} + \cdots + X_{nj}}{n} \tag{2}$$

$$STAD_j = \frac{|X_{1j} - AVG_j| + \cdots + |X_{nj} - AVG_j|}{n} \tag{3}$$

Normalization: The data is uniformly mapped to the interval $[0, 1]$, and N_{ij} is the normalized value of the X_{ij} value, as shown in Equation (4), Equation (5), and Equation (6).

$$N_{ij} = \frac{S_{ij} - X_{\min}}{X_{\max} - X_{\min}} \tag{4}$$

$$X_{\min} = \min\{S_{ij}\} \tag{5}$$

$$X_{\max} = \max\{S_{ij}\} \tag{6}$$

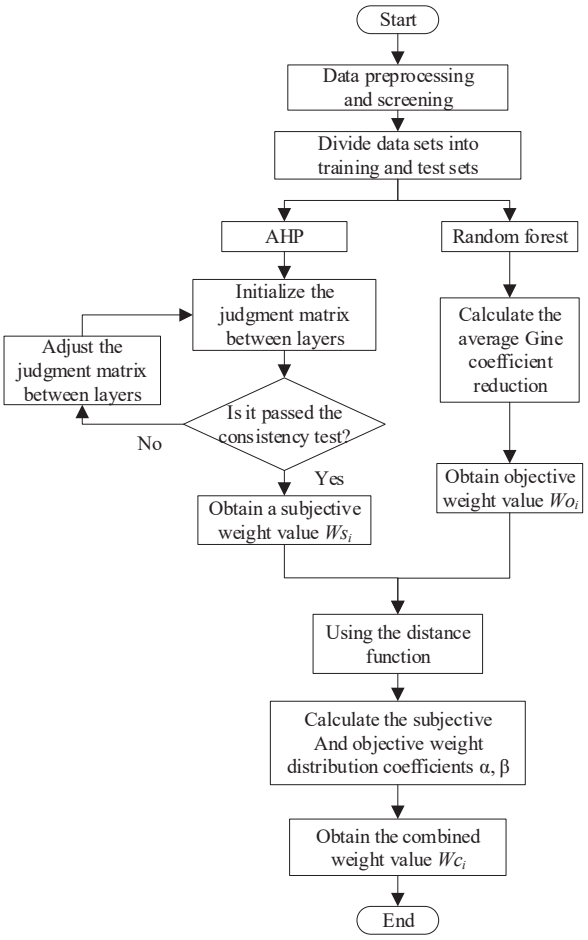


Figure 2: Weight calculation total flow chart

3 Building Malicious Attack Detection Algorithm of IoV based on CW-KNN

3.1 Weight Calculation

The main work of this section is to weight the KNN algorithm using combined weights, the purpose is to get the CW-KNN algorithm. In Part 2, 17 main malicious attack features of the IoV were selected. In this section, combined weights are given to these 17 features. We use the Analytic Hierarchy Process (AHP) to calculate subjective weights, and use random forests to calculate objective weights, then the distance function method is used to calculate combined weights. This not only reflects people's intuitive understanding of malicious attacks, but also reflects the authenticity of objective data, and also can make the results more accurate. The overall calculation process is shown in Figure 2.

(1) Calculating Subjective Weights

The first step is to use AHP to calculate subjective weights. AHP is a decision analysis method that combines qualitative and quantitative methods to solve multi-objective complex problems. It is widely used in various fields.

AHP model is established according to Table 2. As shown in Figure 3. But the established AHP model needs to pass the consistency check [23], details as follows. The calculation method of CI is shown in Equation (7).

$$CI = \frac{\lambda_{\max} - n}{n-1} \tag{7}$$

n is the dimension of the matrix, the value of RI is shown in Table 3.

Table 3: the value of RI

n	1	2	3	4	5	6
RI	0	0	0.58	0.9	1.12	1.24

The consistency ratiois CR, as shown in Equation (8).

$$CR = \frac{CI}{RI} \tag{8}$$

If $CR < 0.1$, passes the consistency check; Begin to calculate the subjective weight of 17 features. The judgment matrix [23] of the Criterion B_j $(j = 1, 2, 3, 4)$ to the Goal A is as shown in Equation (9).

$$A = \begin{bmatrix} 1 & 2 & 2 & \frac{1}{2} \\ \frac{1}{2} & 1 & 1 & \frac{1}{2} \\ \frac{1}{2} & 1 & 1 & \frac{1}{2} \\ 2 & 2 & 2 & 1 \end{bmatrix} \tag{9}$$

The maximum eigenvalue is $\lambda_{\max}$. From the Equation $A\mu = \lambda^*_{\max}\mu$, $\lambda_{\max} = 4.0604$ can be calculated, the eigenvectors of B_j $(j = 1, 2, 3, 4)$ is [0.2775, 0.3925, 0.1650, 0.1650].

$CR = 0.0226 < 0.1$ is calculated. Through the consistency check. The Weight of [B_1, B_2, B_3, B_4] is [0.2775, 0.3925, 0.1650, 0.1650].

The judgment matrix of the sub-criteria C_1-C_5 versus B_1 is as shown in Equation (10).

$$B_1 = \begin{bmatrix} 1 & 1 & \frac{1}{3} & \frac{1}{4} & \frac{1}{4} \\ 1 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 3 & 4 & 1 & \frac{1}{2} & \frac{1}{2} \\ 4 & 4 & 2 & 1 & 1 \\ 4 & 4 & 2 & 1 & 1 \end{bmatrix} \tag{10}$$

$\lambda_{\max} = 5.0552$ of the B_1 can be calculated, and the eigenvectors of C_i $(i = 1, 2, 3, 4, 5)$ is [0. 0751, 0. 0709, 0.2028, 0.3256, 0.3256].

$CR = 0.0123 < 0.1$ is Calculated. Through the consistency check. the weight of [C_1, C_2, C_3, C_4, C_5] is [0. 0751, 0. 0709, 0.2028, 0.3256, 0.3256].

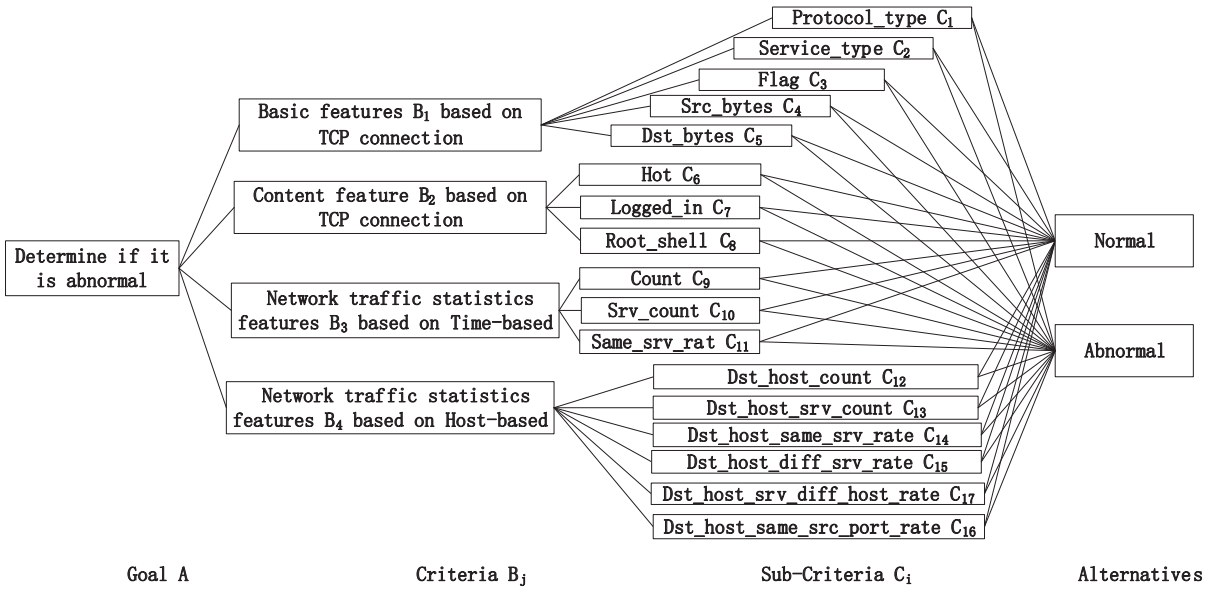


Figure 3: AHP model

The judgment matrix of the sub-criteria C_6-C_8 versus B_2 is as shown in Equation (11).

$$B_2 = \begin{bmatrix} 1 & 6 & 3 \\ \frac{1}{6} & 1 & \frac{1}{3} \\ \frac{1}{3} & 3 & 1 \end{bmatrix} \tag{11}$$

$\lambda_{\max} = 3.0183$ of the B2 can be calculated, and the eigenvectors of C_i $(i = 6, 7, 8)$ is [0.6548, 0. 0953, 0.2499].

$CR = 0.0176 < 0.1$ is Calculated. Through the consistency check. the weight of [C_6, C_7, C_8] is [0.6548, 0. 0953, 0.2499].

The judgment matrix of the sub-criteria C_9-C_{11} versus B_3 is as shown in Equation (12).

$$B_3 = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{2} \\ 3 & 1 & 3 \\ 2 & \frac{1}{3} & 1 \end{bmatrix} \tag{12}$$

$\lambda_{\max} = 3.0536$ of the B_3 can be calculated, and the eigenvectors of C_i $(i = 9, 10, 11)$ is [0.1571, 0.2493, 0.5936].

$CR = 0.0516 < 0.1$ is Calculated. Through the consistency test, the weight of [$C9$, C_{10}, C_{11}] is [0.1571, 0.2493, 0.5936].

The judgment matrix of the sub-criteria C_{12}-C_{17} versus B_4 is as shown in Equation (13).

$$B_4 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & \frac{1}{3} \\ 2 & 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & \frac{1}{2} \\ 3 & 3 & 1 & 2 & 3 & 2 \\ 3 & 3 & \frac{1}{2} & 1 & 2 & 2 \\ 2 & 2 & \frac{1}{3} & \frac{1}{2} & 1 & \frac{1}{3} \\ 3 & 2 & \frac{1}{2} & \frac{1}{2} & 3 & 1 \end{bmatrix} \tag{13}$$

$\lambda_{\max} = 6.2454$ of the B_4 can be calculated, and the eigenvectors of C_i $(i = 12, 13, 14, 15, 16, 17)$ is [0. 0660, 0. 0890, 0.3144, 0.2333, 0.1851, 0.1121].

$CR = 0.0390 < 0.1$ is Calculated. Through the consistency test, the weight of [C_{12}, C_{13}, C_{14}, C_{15}, C_{16}, C_{17}] is [0. 0660, 0. 0890, 0.3144, 0.2333, 0.1851, 0.1121].

The total consistency check of AHP model is as follows.

$$\begin{aligned} CI &= \sum_{j=1}^{4} B_j^* Cl_j \\ &= 0.2775 * \frac{5.0552 - 5}{5 - 1} + 0.3925 * \frac{3.0183 - 3}{3 - 1} \\ &+ 0.1650 * \frac{3.0536 - 3}{3 - 1} + 0.1650 * \frac{6.2454 - 6}{6 - 1} \\ &= 0.0198 \end{aligned}$$

$$\begin{aligned} RI &= \sum_{j=1}^{4} B_j^* RI_j \\ &= 0.2775 * 1.12 + 0.3925 * 0.58 + 0.1650 * 0.58 \\ &+ 0.1650 * 1.24 = 0.83875 \end{aligned}$$

The result is "$CR = CI/RI = 0.0236 < 0.1$", so the total consistency check is passed.

Subjective weight is defined as W_{S_i}. The calculation method of W_{S_i} is shown in Equation (14). And summary in Table 4.

$$W_{S_i} = \begin{cases} c_i * B_1;\ i = 1, 2, 3, 4, 5 \\ c_i * B_2;\ i = 6, 7, 8 \\ c_i * B_3;\ i = 9, 10, 11 \\ c_i * B_4;\ i = 12, 13, 14, 15, 16, 17 \end{cases} \tag{14}$$

(2) Calculation of Objective Weights

The second step uses a random forest to calculate objective weights. Random forests are not prone to overfitting

Table 4: Subjective weights

B layer / c layer	B_1 0.2775	B_2 0.3925	B_3 0.165	B_4 0.165	W_{S_i}
C_1	0. 0751	\	\	\	0. 0208
C_2	0. 0709	\	\	\	0. 0197
C_3	0.2028	\	\	\	0. 0563
C_4	0.3256	\	\	\	0. 0904
C_5	0.3256	\	\	\	0. 0904
C_6	\	0.6548	\	\	0.257
C_7	\	0. 0953	\	\	0. 0374
C_8	\	0.2499	\	\	0. 0981
C_9	\	\	0.1571	\	0. 0259
C_{10}	\	\	0.2493	\	0. 0411
C_{11}	\	\	5936	\	0. 098
C_{12}	\	\	\	0. 066	0. 0109
C_{13}	\	\	\	0. 089	0. 0147
C_{14}	\	\	\	0.3144	0. 0519
C_{15}	\	\	\	0.2333	0. 0385
C_{16}	\	\	\	0.1851	0. 0305
C_{17}	\	\	\	0.1121	0. 0185

and have a high tolerance for outliers and noise. In this paper, the creation of the random forest model is performed in the R Language environment. It can provide some integrated tools, such as the "RandomForest" and "caret" toolkits required for this modeling.

In this paper, an another important reason for choosing a random forest is that the random forest can calculate the importance value of each variable. Random forest provides two basic variable importance values: Mean Decrease Gini and Mean Decrease Accuracy. this paper used Mean Decrease Gini as an objective weight. Some feature weights calculated by the random forest are shown in Figure 4.

```
training finished
 1) count                          0.179618
 2) ecr_i                          0.145816
 3) dst_host_srv_diff_host_rate    0.092629
 4) icmp                           0.071231
 5) same_srv_rate                  0.067123
 6) dst_bytes                      0.056923
 7) udp                            0.055820
 8) dst_host_count                 0.047003
 9) serror_rate                    0.039553
10) srv_count                      0.036703
```

Figure 4: Some features and weights

Objective weight is defined as W_{O_i}, Repeat the experiment 10 times and take the average, The serial number in W_{O_i} corresponds to Table 2. Objective weights calculated by Random Forest as shown in Table 5.

Table 5: Typical states of SEIR model

W_{O_1}	W_{O_2}	W_{O_3}	W_{O_4}
0. 0368	0. 0286	0. 0461	0. 0814
W_{O_5}	W_{O_6}	W_{O_7}	W_{O_8}
0. 0982	0. 0184	0. 0982	0. 0002
W_{O_9}	$W_{O_{10}}$	$W_{O_{11}}$	$W_{O_{12}}$
0.2087	0. 0532	0. 0627	0. 0859
$W_{O_{13}}$	$W_{O_{14}}$	$W_{O_{15}}$	$W_{O_{16}}$
0. 0266	0. 0266	0. 0327	0. 0384
$W_{O_{17}}$			
0. 0573			

(3) Calculation of Combined Weights

The third step uses the distance function method to calculate the combined weight. Because KNN is based on distance, and the distance function method introduces the concept of distance function, therefore, this paper choosed distance function method for combined weighting. The distance function method is used to reduce the difference between subjective and objective weights, so that the subjective and objective weights are organically combined,and this also makes the combination weights statistically significant.

Make W_{C_i} as the combined weight, α is the coefficient of subjective weighting, β is the coefficient of objective weight,as shown in Equation (15).

$$W_{C_i} = \alpha W_{S_i} + \beta W_{O_i} \tag{15}$$

The distance function expressions [10] is shown in Equation (16).

$$\mathrm{d}\left(W_{S_i}, W_{O_i}\right) = \sqrt{\frac{1}{2}\sum_{i=1}^{n}\left(W_{S_i} - W_{O_i}\right)^2} \tag{16}$$

To reduce the difference, make the distribution coefficient equal to the distance function, as shown in Equation (17).

$$\mathrm{d}\left(W_{S_i}, W_{O_i}\right)^2 = (\alpha - \beta)^2 \tag{17}$$

The value of α and β is calculated, as shown in Equation (18). and $\alpha + \beta = 1$.

$$\begin{aligned} \alpha &= \sqrt{\frac{1}{8}\sum_{i=1}^{n}\left(W_{S_i} - W_{O_i}\right)^2} + \frac{1}{2} \\ &= \sqrt{\frac{1}{8} * 0.11368} + \frac{1}{2} \\ &= 0.12 + 0.5 = 0.62 \end{aligned} \tag{18}$$

$\beta = 1 - 0.62 = 0.38$, α and β can be substituted into the Equation (15) to calculate the combination weight of each feature, as shown in Table 6.

3.2 Improve KNN Algorithm

The direct use of KNN in the IoV will reduce the accuracy, because KNN uses Euclidean distance to consider the contribu- tion of all features in the sample as the same, and does not weight features,Therefore, this section is to improve the KNN algorithm. The combined weights calculated in Table 6 are brought into the weighted distance to obtain the CW-KNN classification algorithm. The specific process is as follows.

(1) Weight the Distance

Different features have their corresponding weights. Bring the combined weight W_{C_i} into the Euclidean distance, obtaine the weighted distance of two arbitrary samples x and y, as shown in Equation (19).

$$\mathrm{d}(x, y) = \sqrt{\sum_{i=1}^{n}\left(x_i - y_i\right)^2 W_{C_i}} \tag{19}$$

(2) Building CW-KNN

The main classification decision rule in CW-KNN is a majority vote. The process is as follows.

Algorithm 1 CW-KNN

Input: training dataset $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_i, y_i)\}$; k is the number of neighbors;

Output: The category y to which the instance x belongs;

1: Begin

2: Calculate combination weighted Euclidean distance

$$\mathrm{d}(x, y) = \sqrt{\sum_{i=1}^{n}\left(x_i - y_i\right)^2 W_{C_i}}$$

3: Find the k points closest to x in the training set D,

4: The neighborhood of x covering the k points is denoted as $Nk(x)$

5: In $Nk(x)$, ater majority vote, determine category to which instance x belongs;

6: End

4 Simulation Experiment

4.1 Experimental Benchmarks and Methods

This paper used python3 to perform binary classification experiments on CW-KNN. The experimental benchmark is to use the confusion matrix to analyze from four aspects: Accuracy, Precision, Recall, and F1. In order to verify the efficiency of CW-KNN, it will be compared with many different types of detection methods. Specifically, it includes KNN without combined weighting, SVM(Support Vector Machine) based on machine learning, FCD-KNN [21] based on Related to the Distance of Attribute Values, Adaboost based on ensemble learning and Random Forest based on tree.

The experiment is divided into two parts. The first part is the comparison between CW-KNN and the other two KNN algorithms. The second part is the comparison between CW-KNN and other types of classification algorithms.

Considering the factors of calculation time and memory consumption, in this paper, 10% training set and extracts part of the testing set are finally used for experiments, as shown in Table 7:

4.2 Comparison within KNN

This section reserch on the effect of different values of K on CW-KNN, and compared with the other two KNN algorithms. The value of K is the nearest neighbor number,and it is the most important value in Knn. The value of K will directly affect the quality of classification. The combined weight set by CW-KNN is shown in Table 6. K takes 3 to 10 and $K \in Z$, the experimental results are shown in Figure 5.

From Figure 5 it can be seen that when $K = 7$, the accuracy of all the KNN algorithms is the same. When $k = 8$, the accuracy of FCD-KNN and CW-KNN is the same. When $k \neq 7$ or $\neq 8$, the accuracy of CW-KNN is higher than KNN and FCD-KNN.

In order to reduce the influence of the values of K on experimental results, this paper set $K = 7$, and get the ROC curves of the three kind of KNN algorithms, As shown in Figure 6.

The experiments in this section prove that the accuracy of CW-KNN is higher than KNN and FCD-KNN.

Table 6: Feature combination weight table

Feature number and name $i = 1, 2, \cdots, 17$	Subjective weight W_{S_i}	Objective weight W_{O_i}	Combination weight W_{C_i}
1. protocol_type	0. 0208	0. 0368	0. 0267
2. service	0. 0197	0. 0286	0. 023
3. flag	0. 0563	0. 0461	0. 0524
4. src_bytes	0. 0904	0. 0814	0. 087
5. dst_bytes	0. 0904	0. 0982	0. 0934
6. hot	0.257	0. 0184	0.1663
7. logged_in	0. 0374	0. 0982	0. 0605
8. root_shell	0. 0981	0. 0002	0. 0609
9. count	0. 0259	0.2087	0. 0954
10. srv_count	0. 0411	0. 0532	0. 0457
11. same_srv_rat	0. 098	0. 0627	0. 0846
12. dst_host_count	0. 0109	0. 0859	0. 0394
13. dst_host_srv_count	0. 0147	0. 0266	0. 0192
14. dst_host_same_srv_rate	0. 0519	0. 0266	0. 0423
15. dst_host_diff_srv_rate	0. 0385	0. 0327	0. 0363
16. dst_host_same_src_port_rate	0. 0305	0. 0384	0. 0335
17. dst_host_srv_diff_host_rate	0. 0185	0. 0573	0. 0332

Table 7: Sample distribution of dataset

Num	Type	Number of samples	
		Training	Testing
0	normal	97278	118835
1	abnormal	396743	29371

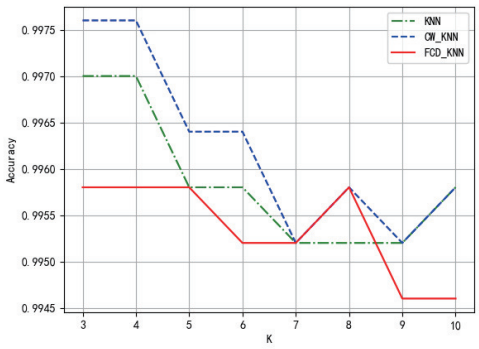


Figure 5: Accuracy with different K values

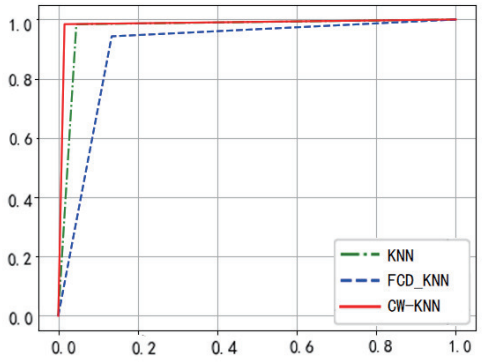


Figure 6: ROC graph of three KNN algorithms

4.3 Comparison Between CW-KNN and Other Classification Algorithms

This section focuses on the measurement of CW-KNN benchmarks,and compared with the other five classification methods.

The value of K of all KNN is set to 7, the other classification algorithm parameters are Python3 original parameters. Obtain the confusion matrix of 6 classification algorithms through experiments, as shown in Tables 8-13.

Comparison of multiple classification results, As shown in Table 14.

From Table 14, it can be seen that the value of F1 of CW-KNN is higher than other classification algorithms, which illustrates CW-KNN is superior in comprehensive performance. Second, CW-KNN has improved in Precision, which shows that CW-KNN has better detection ability than other classification algorithms. however, CW-KNN is inferior to SVM and Adaboost in terms of Accuracy, this is also an issue that needs to be addressed in the next step. In sumary,the experiment proves that the CW-KNN proposed in this paper has better classification effect in binary classification.

Table 8: Confusion matrix of KNN

KNN		prediction	
		normal	abnormal
actual	normal	117910	925
	abnormal	1	29370
Precision		0.994	
Recall		0.992	
Accuracy		0.995	
F1		0.984	

Table 9: Confusion matrix of Random Forest

Random Forest		prediction	
		normal	abnormal
actual	normal	118129	706
	abnormal	18	29353
Precision		0.977	
Recall		0.999	
Accuracy		0.995	
F1		0.988	

Table 10: Confusion matrix of Adaboost

Adaboost		prediction	
		normal	abnormal
actual	normal	118316	519
	abnormal	125	29246
Precision		0.983	
Recall		0.996	
Accuracy		0.996	
F1		0.989	

Table 11: Confusion matrix of FCD-KNN

FCD-KNN		prediction	
		normal	abnormal
actual	normal	118683	152
	abnormal	1	29370
Precision		0.995	
Recall		0.998	
Accuracy		0.995	
F1		0.991	

5 Conclusion

Few researchers currently optimize the classification algorithm for IoV, and the KNN without combined weighting does not consider the difference of sample attribute contribution. Therefore, this paper proposed CW-KNN algorithm for IoV. First of all, we selected the featurs of main threats according to IoV, built a simulated attack dataset of IoV, then calculated the combined weight of each feature, and finally brought the combined weight into the KNN for classification. The experimental results show that the CW-KNN has higher efficiency.

The shortcoming of this paper is that the accuracy of CW-KNN is lower than SVM and Adaboost, this will be the next problem to be solved. With the increase of new types of malicious attacks of IoV, dimensions of data will also increase, KNN is based on distance, so it is not good for multi-dimensional data processing, which may lead to a decline in accuracy. Random forest is better at processing multi-dimensional data, so the next step is to bring the combined weights to the Random Forest for research to improve the accuracy.

Acknowledgement

This research is supported by the National Natural Science Foundations of China under Grants No.61862040, No.61762059 and No. 61762060. The authors gratefully acknow-ledge the anonymous reviewers for their helpful comments and suggestions.

References

[1] S. Adepu and A. Mathur, "From design to invariants: Detecting attacks on cyber physical systems," in *IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C'17)*, pp. 533–540, 2017.

[2] R. C. Baishya, N. Hoque, and D. K. Bhattacharyya, "Ddos attack detection using unique source IP deviation," *International Journal Network Security*, vol. 19, no. 6, pp. 929–939, 2017.

[3] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in iot using SDN," in *The 27th International Telecommunication Networks and Applications Conference (ITNAC'17)*, pp. 1–6, 2017.

[4] J. Y. Chen and X. Z. Xu, "Research on network attack detection based on self-adaptive immune computation," *Computer Science*, vol. 45, no. 6A, pp. 364–370, 2018.

[5] F. Chen, Z. Ye, C. Wang, L. Yan, and R. Wang, "A feature selection approach for network intrusion detection based on tree-seed algorithm and k-nearest neighbor," in *IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS'18)*, pp. 68–72, 2018.

[6] C. Guo, Y. Ping, N. Liu, and S. S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.

[7] T. Jeyaprakash and R. Mukesh, "A survey of mobility models of vehicular adhoc networks and simulators," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 94–101, 2015.

Table 12: Confusion matrix of SVM

SVM		prediction	
		normal	abnormal
actual	normal	118548	287
	abnormal	170	29201
Precision		0.990	
Recall		0.994	
Accuracy		0.997	
F1		0.992	

Table 13: Confusion matrix of CW-KNN

CW-KNN		prediction	
		normal	abnormal
actual	normal	118548	287
	abnormal	170	29201
Precision		0.997	
Recall		0.998	
Accuracy		0.995	
F1		0.993	

[8] X. Jian, W. J. Li, H. Y. Geng, and Y. B. Zhai, "An anti-dos attack rfid security authentication protocol in the internet of vehicles," *Journal of Beijing University of Posts and Telecommunications*, vol. 42, no. 2, pp. 114–119, 2019.

[9] G. Li, "Research on network intrusion detection model based on quantum artificial fish school and fuzzy kernel clustering algorithm," *Software Engineering*, vol. 22, no. 6, pp. 33–37, 2019.

[10] T. H. Li, J. Xue, and X. Wei, "Application of combinedweigh mehod and comprehensive index method based on cask theory in ecological waterway assessment of yangtze river," *Journal of Basic Science and Engineering*, vol. 27, no. 1, pp. 36–49, 2019.

[11] J. P. Liu, W. X. Zhang, and Z. H. Tang, "Adaptive network intrusion detection based on fuzzy rough set-based attribute reduction and gmm-lda-based optimal cluster feature learning," *Control and Decision*, vol. 34, no. 2, pp. 243–251, 2019.

[12] M. R. Mohamed, A. A. Nasr, I. F. Tarrad, and M. Z. Abdulmageed, "Exploiting incremental classifiers for the training of an adaptive intrusion detection model," *International Journal Network Security*, vol. 21, no. 2, pp. 275–289, 2019.

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

[14] K. K. Ravulakollu, Amrita, "A hybrid intrusion detection system: Integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers," *International Journal of Network Security (IJNS'18)*, vol. 20, no. 1, pp. 41–55, 2018.

[15] D. B. Rawat, M. Garuba, L. Chen, and Q. Yang, "On the security of information dissemination in the internet-of-vehicles," *Tsinghua Science and Technology*, vol. 22, no. 4, pp. 437–445,2017.

[16] J. D. Ren, X. Q. Liu, and Q. Wang, "An multi-level intrusion detection method based on KNN outlier detection and random forest," *Journal of Computer Research and Development*, vol. 56, no. 3, pp. 566–575, 2019.

[17] Y. Ren, S. Wang, X. Zhang, and M. S. Hwang, "An efficient batch verifying scheme for detecting illegal signatures," *International Journal Network Security*, vol. 17, no. 4, pp. 463–470, 2015.

[18] T. A. Tchakoucht and M. Ezziyyani, "Building a fast intrusion detection system for high-speed-networks: probe and dos attacks detection," *Procedia Computer Science*, vol. 127, pp. 521–530, 2018.

[19] J. Wang and L. L. Yang, "Multitier ensemble classifiers for malicious network traffic detection," *Journal on Communications*, vol. 39, no. 10, pp. 155–165, 2018.

[20] L. Wei, Y. P. Wang, and H. K. Qin, "An algorithm of the privacy security protection based on location service in internet of vehicles," *Automotive Engineering*, vol. 41, no. 3, pp. 252–258, 2019.

[21] H. H. Xiao and Y. M. Duan, *Improved of KNN Algorithm Based on Related to the Distance of Attribute Values*. PhD thesis, 2013.

[22] P. S. Xie, T. X. Fu, and H. J. Fan, "An algorithm of the privacy security protection based on location service in the internet of vehicles," *International Journal of Network Security*, vol. 21, no. 4, pp. 556–565, 2019.

[23] A. M. Yang and F. Gao, "Cloud computing security evaluation and countermeasure based on AHP-fuzzy comprehensive evaluation," *Journal on Communications*, vol. 37, no. Z1, pp. 104–110, 2016.

[24] H. Zhao, D. Sun, H. Yue, M. Zhao, and S. Cheng, "Dynamic trust model for vehicular cyber-physical systems," *International Journal Network Security*, vol. 20, no. 1, pp. 157–167, 2018.

[25] H. Zhao, H. Yue, T. Gu, and W. Li, "CPS-based reliability enhancement mechanism for vehicular emergency warning system," *International Journal of Intelligent Transportation Systems Research*, vol. 17, no. 3, pp. 232–241, 2019.

Table 14: Comparison of multiple classification results

methods	**Benchmarks**			
	F1	Accuracy	Precision	Recall
CW-KNN	0.993	0.995	0.997	0.998
KNN	0.984	0.995	0.994	0.992
FCD-KNN	0.991	0.995	0.995	0.998
Adaboost	0.989	0.996	0.983	0.996
SVM	0.992	0.997	0.99	0.994
Random Forest	0.988	0.995	0.977	0.999

Biography

Peng-shou Xie was born in Jan.1972. He is a professor and a supervisor of master student at Lanzhou University of Technology. His major research field is Security on Internet of Things. E-mail: xiepsh_lut@163. com

Cheng Fu was born in Jun.1991. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 452708186@qq. com

Tao Feng was born in Dec.1970. He is a professor and a supervisor of Doctoral student at Lanzhou University of Technology. His major research field is modern cryptography theory, network and information security technology. E-mail: fengt@lut. cn

Yan Yan was born in Oct.1980. She is a associate professor and a supervisor of master student at Lanzhou University of Technology. Her major research field is privacy protection, multimedia information security. E-mail: yanyan@lut. cn

Liang-lu Li was born in Jun.1992. She is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 1141685642@qq. com

Anti-SPA Scalar Multiplication Algorithm on Twisted Edwards Elliptic Curve

Shuang-Gen Liu, Xin Heng, and Yuan-Meng 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 May 9, 2019; Revised and Accepted Nov. 6, 2019; First Online Feb. 1, 2020)

Abstract

In order to improve the efficiency of scalar multiplication on the Twisted Edwards curve, the mathematical formula is used to optimize the equations, and the new point addition, double point and point tripling calculation formulas are obtained, which makes the calculation efficiency increase by 24.0%, 24.8% and 22.7% respectively compared with the original calculation formula. Based on balanced ternary, a new round-down balanced ternary scalar multiplication algorithm against SPA attacks was proposed, and combined with the Twisted Edwards curve characteristics. When the ternary scalar length is 101 bits, the computational efficiency are improved by 13.5%, 26.3% and 26.6% compared with the BTSM algorithm, the STF algorithm and the HSTF algorithm, respectively.

Keywords: Balanced Ternary; Elliptic Curve Cryptosystem; Scalar Multiplication; Twisted Edwards Curve

1 Introduction

In 1985, Miller [**?**] and Koblitz [**?**] proposed the elliptic curve cryptography (ECC), which has become one of the research hot spots in the field of cryptography in recent years. Because ECC can achieve the same security requirements as the a cryptosystem based on finite field with a shorter key length and thus can complete encryption and decryption operations at a faster speed . At present, ECC has been widely used in cryptographic chips, e-commerce, wireless communications, satellite communications and other fields. The development of Internet technology makes the radio frequency identification (RFID) rendering large-scale application requirements, as a result, the RFID protocols based on ECC encryption also arises at the historic moment.

The most basic and time-consuming operation in the elliptic curve cryptosystem is the scalar multiplication algorithm kP, where k is an integer, P is a point defined on the elliptic curve E on the field Fq, and $kP = P + ... + P$. It determines the operation speed of the elliptic curve cryptosystem. When the two points on the elliptic curve are the same, the sum is called the double point operation. If not same, it is called the point addition operation. Scalar multiplication algorithm includes basic operations such as field multiplication, field addition, field square and inverse, among which the most expensive operation is inverse operation.

There are many ways to improve the efficiency of the elliptic curve scalar multiplication algorithm, in which different expansion forms of the scalar k can be studied, thereby reducing the number of point addition or double point in the scalar multiplication, such as binary expansion,ternary expansion [**?**, **?**, **?**], w-NAF [**?**], addition Chain [**?**], and other forms to represent scalar k. Or can reduce the field multiplication in the formula, the number of field square calculations [**?**], or converted the inverse operation to field square or field multiplication [**?**] to optimize the point addition and double point operation.In addition, the expansion of k can be combined with different elliptic curves. In [**?**] and [**?**], a formula for $3P$ operation in Jacobian coordinate and the ternary Montgomery algorithm on Hessian curve were proposed respectively, all of those algorithms improve the operational efficiency of the scalar multiplication.

The remainder of the paper is structured as follows: Section 2 introduces the basics of the Twisted Edwards curve and balanced ternary [**?**]. Section 3 proposes an elliptic curve scalar multiplication optimization algorithm on the Twisted Edwards curve. Section 4 provides efficiency analysis and comparison with other algorithms. Finally, the Section 5 draws a conclusion.

2 Basis Knowledge

2.1 Twisted Edwards Curve

In 2007, Edwards proposed a new representation form of elliptic curve that field feature is not 2, which is called the elliptic curve of Edwards form and referred to as Edwards

curve for short [?]. It is defined as:

$$x^2 + y^2 = c^2(1 + dx^2y^2)$$

Where $c, d \in k$, $c, d \neq 0$, and $cd^4 \neq 1$. The Edwards curve has simple group operation rules, unified point addition and double point formula, and the efficiency and safety of the algorithm are higher than those of Hessian, Jacobian, Doche, *etc.* [?]. Bernstein improved the Edwards curve in 2008. He proposed the Twisted Edwards curve and proved that it covered more curves on the finite field than the Edwards form [?, ?, ?], and the basic computational efficiency of the Twisted Edwards curve is higher.In recent years, more and more research has been done on the twisted Edwards curve. in [?], an ECDSA signature for double scalar multiplication on a distorted Edwards curve is proposed. [?] propose formulas for computing 3 and 4 isogenies on twisted Edwards curves and [?] propose a coordinate system for elliptic curve cryptosystem on twisted Edwards curve. These studies have made Twisted Edwards more widely used.

Let E be the field whose feature value is not 2. $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ are two points on Twisted Edwards curve, the curve equation is:

$$E : ax^2 + y^2 = 1 + dx^2y^2 \tag{1}$$

The basic operation rules [?] are as follows:

1) Unit: For all $P \in E$, $P + \infty = \infty + P$;
2) Negative: If $P = (x, y) \in E$, then $(x, y) + (-x, y) = \infty$. Let the negative of P be $-P = (-x, y)$.

Point addition operation: Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, $P_1, P_2 \in E$ and $P_1 \neq P_2$, then $P_1 + P_2 = P_3 = (x_3, y_3)$.

$$\begin{cases} x_3 = \dfrac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2} \\ y_3 = \dfrac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2} \end{cases} \tag{2}$$

Double point operation: Let $P_1 = (x_1, y_1)$, $P_1 \in E$ then $2P_1 = P_4 = (x_4, y_4)$.

$$\begin{cases} x_4 = \dfrac{2x_1y_1}{1 + dx_1^2y_1^2} \\ y_4 = \dfrac{y_1^2 - ax_1^2}{1 - dx_1^2y_1^2} \end{cases} \tag{3}$$

Can be introduced by Equation (**??**):

$$1 + dx^2y^2 = ax^2y^2 \tag{4}$$

$$1 - dx^2y^2 = 2 - (ax^2 + y^2) \tag{5}$$

Bring Equation (**??**) and Equation (**??**) into Equation (**??**), at which double point formula becomes:

$$\begin{cases} x_4 = \dfrac{2x_1y_1}{ax_1^2 + y_1^2} \\ y_4 = \dfrac{y_1^2 - ax_1^2}{2 - (ax_1^2 + y_1^2)} \end{cases} \tag{6}$$

Equation (**??**) reduces the one field multiplication operation compared to Equation (**??**).

Point tripling operation:

Let $P_1 = P_2 = P_3 = (x_1, y_1)$, $P_1 + P_2 + P_3 = 3P_1 = (x_5, y_5)$. Bring the double point formula into the point addition formula, which can be obtained from Equation (**??**):

$$d = \frac{ax^2 + y^2 - 1}{x^2y^2}$$

Instead of the d in the point addition formula, point tripling operation formula can be obtained:

$$\begin{cases} x_5 = \dfrac{(ax_1^2 + y_1^2)^2 - 4y_1^2}{4ax_1^2(ax_1^2 - 1)^2 - (ax_1^2 + y_1^2)^2}x_1 \\ y_5 = \dfrac{(ax_1^2 + y_1^2)^2 - 4ax_1^2}{4y_1^2(y_1^2 - 1)^2 - (ax_1^2 + y_1^2)^2}y_1 \end{cases} \tag{7}$$

In the underlying field operations of the above mentioned point addition, double point, and point tripling, let I denote the inversion operation, S denote the field square operation, and M denote the field multiplication operation, generally $I = 10M$, $S = 0.8M$. Then, the point addition operation amount is $2I + 5M = 25M$, the double point is $2I + 1M + 3S = 23.4M$, and the triple point is $2I + 6S + 6M = 30.8M$.

2.2 Balanced Ternary

For any integer k, it can be expressed as $k = a_na_{n-1}a_1a_0$, where $a_n = 1$, a_{n-1}, ... , a_1, a_0 is a form of any one of -1,0,1, called the balanced ternary form(BTF) [?]. For balanced ternary, 160-bit and 256-bit binary correspond to 101-bit and 162-bit ternary, respectively [?]. The balanced ternary form expansion algorithm is shown in Algorithm **??**, and balanced ternary scalar multiplication (BTSM) algorithm is given by Algorithm **??**.

Algorithm 1 Balanced ternary expansion algorithm

Input: integer k
Output:$k = (k_{n-1}k_{n-2}\cdots k_1k_0)_3, k_i \in \{0, 1, -1\}$
$i \leftarrow 0$
while $k > 0$ **do**
  if k *mod* 3==2 **then**
    $k_i \leftarrow -1$;
    $k = \lceil k/3 \rceil$;
  else if k *mod* 3==1 **then**
    $k_i \leftarrow 1$;
    $k = \lfloor k/3 \rfloor$;
  else
    $k_i \leftarrow 0$;
    $k = k/3$;
  end if
  $i \leftarrow i + 1$;
end while
Return $k = (k_{n-1}k_{n-2}\cdots k_1k_0)_3$
End

Algorithm 2 Balanced ternary scalar multiplication algorithm(BTSM)

Input: $k = (k_{n-1}k_{n-2}\cdots k_1k_0)_3, P$
Output:kP
$Q \leftarrow O$
for n-1 to 0,$i--$ **do**
  $Q \leftarrow 3Q$;
  if $k_i = 1$ **then**
    $Q \leftarrow Q + P$;
  else if $k_i = -1$ **then**
    $Q \leftarrow Q - P$;
  end if
end for
Return Q
End

Algorithm **??** performs a point tripling operation every time. Only when k_i=1 or -1, a point addition operation is performed. The probability of occurrence of 1 and -1 is 2/3. In the operation, A represents a point addition operation, D represents a double point operation, and T represents a point tripling operation. So the total calculation amount require $nT + (2/3)nA$. In [**?**], a anti-SPA algorithm based on ternary is proposed, the STF algorithm and the HSTF algorithm require $nT+nD$ and $nT + 1D + nA$ respectively.

3 New Algorithm on the Twisted Edwards Curve

3.1 Optimization Algorithm of Field Operation on The Twisted Edwards Curve

The basic part of the previous section mentioned the operation rules of the Twisted Edwards curve. The operations of the underlying field operations of point addition, double point and point tripling are $2I + 5M = 25M$, $2I + 1M + 3S = 23.4M$ and $2I + 6S + 6M = 30.8M$ respectively. It can be seen that each of operations costs two inverse that make the overall computational complexity too large. Therefore, the mathematical formulas (10) and (11) are used to propose new computational formulas for algorithm optimization.

$$\begin{cases} A^{-1} = (AB)^{-1}B \\ B^{-1} = (AB)^{-1}A \end{cases} \tag{8}$$

$$A_1B_2 + A_2B_1 = (A_1 + B_1)(A_2 + B_2) - A_1A_2 - B_1B_2 \tag{9}$$

By applying Equation (**??**) and Equation (**??**) to Equation (**??**), the optimized point addition formula can be obtained:

$$\begin{cases} x_3 = \dfrac{(x_1 + y_1)(x_2 + y_2) - x_1x_2 - y_1y_2}{(1 + dx_1x_2y_1y_2)(1 - dx_1x_2y_1y_2)} \\ \qquad \cdot(1 - dx_1x_2y_1y_2) \\ y_3 = \dfrac{y_1y_2 - x_1x_2}{(1 - dx_1x_2y_1y_2)(1 + dx_1x_2y_1y_2)} \\ \qquad \cdot(1 + dx_1x_2y_1y_2) \end{cases}$$

The Equation (**??**) applied to Equation (**??**), can be optimized double point formula:

$$\begin{cases} x_4 = \dfrac{2x_1y_1[2 - (ax_1^2 + y_1^2)]}{(ax_1^2 + y_1^2)[2 - (ax_1^2 + y_1^2)]} \\ y_4 = \dfrac{(y_1^2 - ax_1^2)(ax_1^2 + y_1^2)}{[2 - (ax_1^2 + y_1^2)](ax_1^2 + y_1^2)} \end{cases}$$

By applying Equation (**??**) to Equation (**??**), the optimized point tripling formula can be obtained:

$$\begin{cases} x_5 = \dfrac{[(ax_1^2 + y_1^2)^2 - 4y_1^2]x_1}{[4ax_1^2(ax_1^2 - 1)^2 - (ax_1^2 + y_1^2)^2]} \\ \qquad \dfrac{\cdot[4y_1^2(y_1^2 - 1)^2 + (ax_1^2 + y_1^2)^2]}{\cdot[4y_1^2(y_1^2 - 1)^2 + (ax_1^2 + y_1^2)^2]} \\ y_5 = \dfrac{[(ax_1^2 + y_1^2)^2 - 4ax_1^2]y_1}{4y_1^2(y_1^2 - 1)^2 + (ax_1^2 + y_1^2)^2} \\ \qquad \dfrac{\cdot[4ax_1^2(ax_1^2 - 1)^2 - (ax_1^2 + y_1^2)^2]}{\cdot[4ax_1^2(ax_1^2 - 1)^2 - (ax_1^2 + y_1^2)^2]} \end{cases} \tag{10}$$

The optimized point addition, double point, and point tripling formulas makes the denominator of the calculation formula of x and y unified by the Equation (**??**), thus reducing the inverse operation, adding the three field multiplication operation. At the same time, the use of Equation (**??**) for point addition operation reduces the one field multiplication operation. The calculated operations of point addition, double point, and point tripling are $1I + 9M = 19M$, $1I + 6M + 2S = 17.6M$ and $1I + 6S + 9M = 23.8M$ respectively.

3.2 Anti-SPA Round-down Symmetric Ternary Scalar Multiplication Algorithm

Simple power analysis (SPA) attack is one of the security threats of elliptic curve cryptosystems. Because the formulas for calculating the underlying field of point addition and double point are very different, the energy trajectory can be distinguished directly. Therefore, based on the measured power consumption trajectory, the attacker judges the program and operation input by the encryption device according to a certain moment, thereby recovering the currently used key information [**?**]. Any implementation that determines the execution route by the key bit is potentially vulnerable to be attacked. In order to resist SPA attacks, a new scalar multiplication algorithm uses an uniform point addition and double point formula. The algorithm are as Algorithm **??** and Algorithm **??**.

Algorithm 3 Round-down BTF expansion algorithm

Input: integer k
Output: $(k_{n-1}, k_{n-2}, ..., k_1, k_0)_3$
$i \leftarrow 0$
while $k > 0$ **do**
  if $k \ mod \ 3 = 2$ **then**
    $k_i \leftarrow -1, k \leftarrow \lfloor k/3 \rfloor$
  else if $k \ mod \ 3 = 1$ **then**
    $k_i \leftarrow 1, k \leftarrow \lfloor k/3 \rfloor$
  else
    $k_i \leftarrow 0, k \leftarrow k/3$
  end if
  $i \leftarrow i + 1$
end while
Return $(k_{n-1}, k_{n-2}, ..., k_1, k_0)_3$
End

Algorithm 4 Anti-SPA round-down BTSM algorithm

Input: integer k, $P \in E(F_q)$
Output: $Q = kP$
Use algorithm 3 to represent k as round-down BTF: $k = (k_{n-1}, k_{n-2}, ..., k_1, k_0)_3$
$T_1 \leftarrow O, T_2 \leftarrow P$
$i = 1$
for 0 *to* $n-2$, $j++$ **do**
  if $k_j = 0$ **then**
    $T_{i+3} = 2T_i + T_{i+1}, T_{i+2} = T_{i+3} - P$
  else if $k_j = 1$ **then**
    $T_{i+2} = 2T_i + T_{i+1}, T_{i+3} = T_{i+2} + P$
  else if $k_j = -1$ **then**
    $T_{i+2} = 2T_{i+1} + T_i, T_{i+3} = T_{i+2} + P$
  end if
  $i = i + 2$
end for
if $k_j = 0$ **then**
  $T_{i+2} = 2T_i + T_i, T_{i+3} = T_{i+2} + P$
else if $k_j = 1$ **then**
  $T_{i+2} = 2T_i + T_{i+1}, T_{i+3} = T_{i+2} + P$
else if $k_j = -1$ **then**
  $T_{i+2} = 2T_{i+1} + T_i, T_{i+3} = T_{i+2} + P$
end if
Return T_{i+2}
End

Algorithm 5 Anti-SPA round-down BTSM algorithm on the Twisted Edwards curve

Input: integer k, $P = (x_1, y_1) \in E(F_q)$
Output: $Q = kP$
Use algorithm 3 to represent k as round-down BTF: $k = (k_{n-1}, k_{n-2}, ..., k_1, k_0)_3$
$T_1 \leftarrow O, T_2 \leftarrow P$
$A_1 \leftarrow 0, B_1 \leftarrow 0, C_1 \leftarrow 0, D_1 \leftarrow 0, A_2 \leftarrow x_1, B_2 \leftarrow 0, C_2 \leftarrow y_1, D_2 \leftarrow 0$
$T_i = (A_i, B_i, C_i, D_i)$
$T_M, T_N, T_P, T_Q, T_R \leftarrow (0, 0, 0, 0)$
$i = 1$
for 0 to $n-2$, $j++$, combined with Algorithm **??** **do**
  if $k_j = 0$ **then**
    $T_M \leftarrow T_i,\ T_N \leftarrow T_{i+1},\ T_P \leftarrow 2T_M,\ T_Q \leftarrow T_P + T_N,\ T_R \leftarrow T_Q - P,\ T_{i+2} \leftarrow T_R,\ T_{i+3} \leftarrow T_Q$
  else if $k_j = 1$ **then**
    $T_M \leftarrow T_i,\ T_N \leftarrow T_{i+1},\ T_P \leftarrow 2T_M,\ T_Q \leftarrow T_P + T_N,\ T_R \leftarrow T_Q + P,\ T_{i+2} \leftarrow T_Q,\ T_{i+3} \leftarrow T_R$
  else if $k_j = -1$ **then**
    $T_M \leftarrow T_{i+1},\ T_N \leftarrow T_i,\ T_P \leftarrow 2T_M,\ T_Q \leftarrow T_P + T_N,\ T_R \leftarrow T_Q + P,\ T_{i+2} \leftarrow T_Q,\ T_{i+3} \leftarrow T_R$
  end if
  $i = i + 2$
end for
if $k_{n-1} = 0$ **then**
  $T_M \leftarrow T_i,\ T_N \leftarrow T_i,\ T_P \leftarrow 2T_M,\ T_Q \leftarrow T_P + T_N,\ T_R \leftarrow T_Q - P,\ T_{i+2} \leftarrow T_R,\ T_{i+3} \leftarrow T_Q$
else if $k_{n-1} = 1$ **then**
  $T_M \leftarrow T_i,\ T_N \leftarrow T_{i+1},\ T_P \leftarrow 2T_M,\ T_Q \leftarrow T_P + T_N,\ T_R \leftarrow T_Q + P,\ T_{i+2} \leftarrow T_Q,\ T_{i+3} \leftarrow T_R$
else if $k_{n-1} = -1$ **then**
  $T_M \leftarrow T_{i+1},\ T_N \leftarrow T_i,\ T_P \leftarrow 2T_M,\ T_Q \leftarrow T_P + T_N,\ T_R \leftarrow T_Q + P,\ T_{i+2} \leftarrow T_Q,\ T_{i+3} \leftarrow T_R$
end if
if $T_{i+2} = T_R$ **then**
  Return $(\frac{A_R D_R}{C_R D_R}, \frac{C_R B_R}{B_R D_R})$
else if $T_{i+2} = T_Q$ **then**
  Return $(\frac{A_Q D_Q}{C_Q D_Q}, \frac{C_Q B_Q}{B_Q D_Q})$
end if
End

Algorithm **??** adopts a unified double - addition - addition formula. The attacker can not obtain the key information by executing the difference in the branch statement, but the operation efficiency of the algorithm is reduced. Therefore, Algorithm **??** is combined with the Twisted Edwards curve which optimize the underlying field operations. The specific algorithm is as Algorithm **??**.

The meaning of the arrows in Steps 6 and 7 of the Algorithm **??** is to assign the value of $T(A, B, C, D)$ on the right side of the arrow to the left side of the arrow. The underlying field operation solution for T_P,T_Q, and T_R is as Algorithm **??**.

Table 1: T_P, T_Q, T_R specific operation costs

T_i	**i=P**	**i=Q**	**i=R**
A_i	$3M$	$6M$	$4M$
B_i	$2S$	$6M$	$4M$
C_i	0	$2M$	$2M$
D_i	$1M + 1S$	0	$1M$
Total costs	$4M + 3S$	$14M$	$11M$

Table 1 intuitively shows the costs of Algorithm **??** in

Algorithm 6 T_P, T_Q, T_R underlying field operations

Input: $T_M = (A_M, B_M, C_M, D_M)$, $T_i = (A_i, B_i, C_i, D_i)$
Output: $T_P = (A_P, B_P, C_P, D_P)$, $T_Q = (A_Q, B_Q, C_Q, D_Q)$, $T_R = (A_R, B_R, C_R, D_R)$
$A \leftarrow A_M D_M, B \leftarrow B_M C_M, C \leftarrow B_M D_M, D \leftarrow A^2, E \leftarrow B^2, F \leftarrow C^2$
$A_P \leftarrow 2AB, B_P \leftarrow aD + E, C_P \leftarrow E - aD, D_P \leftarrow 2F - B_P$
$A' \leftarrow A_P D_P, B' \leftarrow B_P C_P, C' \leftarrow B_P D_P, D' \leftarrow A_P C_P$
$E' \leftarrow A_N D_N, F' \leftarrow B_N C_N, G' \leftarrow A_N C_N, H' \leftarrow B_N D_N$
$G \leftarrow A'F', H \leftarrow B'E', I \leftarrow C'H', J \leftarrow D'G', K \leftarrow B'F', L \leftarrow A'E'$
$A_Q \leftarrow G + H, B_Q \leftarrow I + dJ, C_Q \leftarrow K - aL, D_Q \leftarrow I - dJ$
$A'' \leftarrow A_Q D_Q, B'' \leftarrow B_Q C_Q, C'' \leftarrow A_Q C_Q, D'' \leftarrow B_Q D_Q$
$G' \leftarrow A''y_1, H' \leftarrow B''x_1, I' \leftarrow x_1 y_1, J' \leftarrow I'D'', K' \leftarrow B''y_1, L' \leftarrow A''x_1$
if $T_R = T_P + P = (A_R, B_R, C_R, D_R)$ **then**
  $A_R \leftarrow G' + H', B_R \leftarrow C'' + dJ', C_R \leftarrow K' - aL', D_R \leftarrow C'' - dJ'$
else if $T_R = T_P - P = (A_R, B_R, C_R, D_R)$ **then**
  $A_R \leftarrow G' - H', B_R \leftarrow C'' - dJ', C_R \leftarrow K' + aL', D_R \leftarrow C'' + dJ'$
end if
Return $(A_P, B_P, C_P, D_P), (A_Q, B_Q, C_Q, D_Q), (A_R, B_R, C_R, D_R)$
End

calculating T_P, T_Q, T_R, and thus the total amount of computation of Algorithm 5 can be directly calculated from the data of Table 1. Algorithm **??** does not perform an inverse operation when performing the underlying field operation of point addition and double point. Only when the value in the last step is returned, does an inverse operation performed using Equation (**??**). It has an unified scalar multiplication formula that can resist SPA attacks and improve computational efficiency.

4 Efficiency Analysis

The optimization algorithm of the underlying field operation of the Twisted Edwards curve proposed in Section 3.1 reduces the underlying field operation of the original Twisted Edwards curve. Table 2 intuitively gives the original and optimized operations of point addition, double point, and point tripling. In comparison, the efficiency of the optimized point addition, double point, and point tripling are 24.0%, 24.8%, and 22.7% higher than the original operation efficiency.

When the Algorithm **??** executes a conditional statement, by combining the underlying field operations of the Algorithm **??**, the inverse operations of the point addition and double point execution are not required, and only the T_P, T_Q, and T_R operations are performed n times, an inverse operation and three field multiplication operation are performed when the algorithm return the value. It is concluded by Algorithm **??** that T_P requires $4M + 3S$, T_Q requires $14M$, and T_R requires $11M$, so the computational amount of Algorithm **??** is $(29M+3S)n+1I+3M$. The computations of the BTSM algorithm, the STF algorithm and the HSTF algorithm using the underlying operation of the Twisted Edwards curve are $(1I + 6S + 9M)n + (1I + 9M)(2/3)n$, $(1I + 6S + 9M)n + (1I + 9M)n$ and $(1I + 6S + 9M)n + (1I + 9M)n + 1I + 2S + 6M$ respectively. Table 3 visually shows the comparison of the computational costs of several algorithms at 101-bit and 162-bit lengths, respectively.

5 Conclusions

In this paper, the two inversions in the original calculation formula of the underlying field operations of the point addition, double point and point tripling on the Twisted Edwards curve are converted into one inversion and three field multiplication. The formula $A_1B_2 + A_2B_1 = (A_1 + B_1)(A_2 + B_2) - A_1A_2 - B_1B_2$ makes the point addition operation reduce the domain multiplication by one time. Therefore, the calculation efficiency of point addition, double point, and point tripling are 24.0%, 24.8%, and 22.7% higher than the original calculation efficiency. In addition, this paper proposes Anti-SPA round-down BTSM algorithm on the Twisted Edwards curve, and combines the characteristics of Twisted Edwards curve, so that the inverse operation is not needed in the whole operation process, only when the last step returns a value does an inverse operation perform. Therefore, when the scalar length is 101 bits, the computational efficiency are 13.5%, 26.3%, and 26.6% higher than that of the BTSM algorithm, the STF algorithm, and the HSTF algorithm respectively. When the scalar length is 162 bits, the computational efficiency is improved by 13.7% , 26.4% and 26.6% respectively.

Acknowledgments

The support of NSFC (National Natural Science Foundation of China, No.61872058), Shaanxi Natural Science Foundation (No.2017JQ6010) is gratefully acknowledged.

References

[1] J. D. Bernstein and T. Lange, "Fast addition and doubling on elliptic curves," in *Proceedings of ASIACRYPT*, pp. 29-50, 2007.
[2] W. Y. Deng and X. H. Miao, "Application of balanced ternary in elliptic curve scalar multiplication," *Computer Engineering*, vol. 38, no. 5, pp. 152–154, 2012.

Table 2: Comparison of the calculation of the original field calculation and optimized on the Twisted Edwards curve

Operation	Original field calculation	Optimized field calculation
Point addition$(P+Q)$	$2I+5M$	$1I+9M$
Double point$(2P)$	$2I+1M+3S$	$1I+6M+2S$
Point tripling$(3P)$	$2I+6S+6M$	$1I+6S+9M$

Table 3: Computation cost of different scalar multiplication algorithms

Algorithm	Anti-SPA	Total costs	101bit	162bit
BTSM Algorithm	No	$(1I+6S+9M)n+(1I+9M)(2/3)n$	$3683M$	$5908M$
STF Algorithm	Yes	$(1I+6S+9M)n+(1I+9M)n$	$4323M$	$6934M$
HSTF Algorithm	Yes	$(1I+6S+9M)n+(1I+9M)n+1I+2S+6M$	$4340M$	$6951M$
Algorithm 5	Yes	$(29M+3S)n+1I+3M$	$3184M$	$5100M$

[3] H. M. Edwards, "A normal form for elliptic curves," *Bulletin of the American Mathematical Society (1979-present)*, vol. 44, no. 3, pp. 393–422, 2007.

[4] K. Eisentrager, K. Lauter, and P. Montgomery, "Fast elliptic curve arithmetic and improved weil pairing evaluation," in *RSA Conference, Cryptographers' Track*, pp. 343–354, 2003.

[5] R. R. Goundar, Joye, Marc, and Miyaji, "Co_Z addition formula and binary ladders on elliptic curves," in *Proceedings of the 12th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'10)*, pp. 65–79, 2010.

[6] H. Hisll, K. Wong, and G. Garter, "Jacobi quartic curve revisited," in *Information Security and Privacy*, vol. 6, no. 25, pp. 452–468, 2009.

[7] T. Izu and T. Takagi, "A fast parallel elliptic curve multiplication resistant against side channel attacks," in *International Workshop on Practice and Theory in Public Key Cryptography*, pp. 371–374., 2002.

[8] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[9] 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 (PIERS - FALL)*, pp. 879–885, 2017.

[10] L. H. Liu and Y. Shen, "Fast algorithm for scalar multiplication in elliptic curves cryptography," *Application Research of Computers*, vol. 26, no. 3, pp. 1104–1105, 2009.

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

[12] S. G. Liu, H. Y. Ni, and Y. P. Hu, "A new kind of elliptic curve scalar multiplication algorithm resistant to power attacks," *Journal of Wuhan University of Technology*, vol. 32, no. 07, pp. 156–159, 2010.

[13] S. G. Liu, G. L. Qi, and X. A. Wang, "Fast and secure elliptic curve scalar multiplication algorithm based on a kind of deformed fibonacci-type series," in *The 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 398–402, Nov. 2015.

[14] S. G. Liu, R. R. Wang, and S. Y. Li, "Ternary montgomery algorithm on hessian curve over GF (3m)," *Journal of Shandong University (Natural Science)*, vol. 54, no. 01, pp. 96–102, 2019.

[15] S. G. Liu, H. T. Yao, and X. A. Wang, "SPA resistant scalar multiplication based on addition and tripling indistinguishable on elliptic curve cryptosystem," in *The 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 785–790, Nov. 2015.

[16] M. Shirase, "Coordinate system for elliptic curve cryptosystem on twisted edwards curve," in *IEEE International Conference on Consumer Electronics-Taiwan*, May 2016.

[17] V. S. Miller, "Use of elliptic curves in cryptography," *Lecture Notes in Computer Science Springer Verlag*, vol. 218, pp. 417–426, 1986.

[18] S. Kim, K. Yoon, and J. Kwon, "Efficient isogeny computations on twisted edwards curves," *Security and Communication Networks*, vol. 66, no. 5, 2018.

[19] D. Vassil, I. Laurent, and K. Pradeep, "Efficient and secure elliptic curve point multiplication using double-base chains," in *Annual International Conference on the Theory and Application of Crytology and Information Security*, pp. 59–78, 2005.

[20] Z. Liu, J. Grossschadl, and Z. Hu, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things," *IEEE Tansactions on Computers*, vol. 66, no. 5, pp. 773–785, 2017.

[21] M. Zhong, H. H. Jia, and L. Y. Jiang, "The optimization of dpa defense system based on quantum annealing algorithm," *Netinfo Security (in Chinese)*, vol. 16, no. 3, pp. 28–33, 2016.

[22] M. Zhou and H. B. Zhou, "Optimization of fast point multiplication algorithm based on elliptic curve," *Application Research of Computers*, vol. 29, no. 08, pp. 3056–3058, 2012.

Biography

Shuang-Gen Liu, born in 1979, Ph.D, associate professor. A member of the China Computer Federation, and a member of the Chinese Association for Cryptologic Research. His main research interests focus on information security and cryptography.

Xin Heng, 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.

Yuan-Meng Li, born in 1997. An undergraduate student in information security in Xi'an University of posts and telecommunications. Her main research interest is information security.

Classifying Malware Images with Convolutional Neural Network Models

Ahmed Bensaoud, Nawaf Abudawaood, and Jugal Kalita
(Corresponding author: Ahmed Bensaoud)

Department of Computer Science, University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy, Colorado Springs, CO 80918, USA
(Email: abensaou@uccs.edu)
(Received Dec. 27, 2019; Revised and Accepted May 23, 2020)

Abstract

Due to increasing threats from malicious software (malware) in both number and complexity, researchers have developed approaches to automatic detection and classification of malware, instead of analyzing methods for malware files manually in a time-consuming effort. At the same time, malware authors have developed techniques to evade signature-based detection techniques used by antivirus companies. Most recently, deep learning is being used in malware classification to solve this issue. In this paper, we use several convolutional neural network (CNN) models for static malware classification. In particular, we use six deep learning models, three of which are past winners of the ImageNet Large-Scale Visual Recognition Challenge. The other three models are CNN-SVM, GRU-SVM and MLP-SVM, which enhance neural models with support vector machines (SVM). We perform experiments using the Malimg dataset, which has malware images that were converted from Portable Executable malware binaries. The dataset is divided into 25 malware families. Comparisons show that the Inception V3 model achieves a test accuracy of 99.24%, which is better than the accuracy of 98.52% achieved by the current state of the art system called the M-CNN model.

Keywords: Convolutional Neural Network; Malware Classification; Malware Detection; ImageNet

1 Introduction

Internet connectivity is an essential infrastructure for business organizations, banking institutions, universities, and governments, and is growing exponentially. This growth is threatened by attackers with malicious codes and network threats [41]. The execution of malware forces a computer to perform operations that are not normal, and may harm a victim's computer systems. The amount of malware in circulation has been increasing rapidly in the recent years, and malware has affected computer systems all over the world [21]. Thousands of malware files are being created daily. Figure 1 presents annual statistics of malware attacks over the last 10 years, showing that the total number of malware in circulation has increased to more than 900 million in 2019, which is a 2000% increase compared to the number of malware in the year 2010 [26].

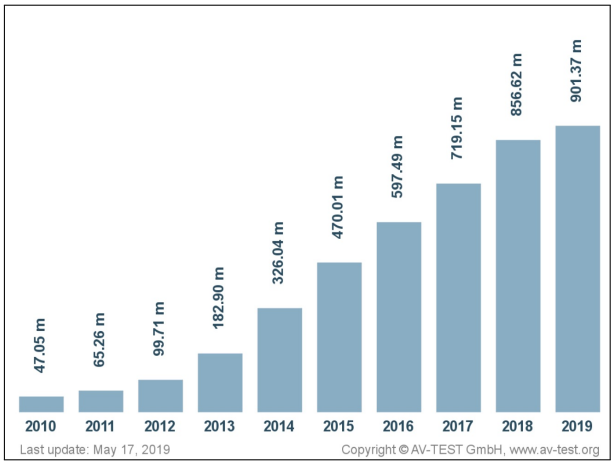


Figure 1: Number of worldwide malware attacks for the last ten years [26].

The cost of malware infection can run into millions of dollars for each incident inflicted upon small and medium sized businesses [32]. Routing protocols alone are not sufficient to detect malware [48]. As a result, researchers and anti-virus vendors employ machine learning to detect and classify malicious software. A large number of studies have focused on malware binary since binaries are normally used to infect computers. Malware is analyzed based on static as well as dynamic analysis. While static analysis extracts malware features that can be used to detect or classify malware employing machine learning, dynamic analysis analyzes malware behavior as it is executed in a controlled environment like Cuckoo Sandbox [14], which is open source, available on GitHub.

Various traditional machine learning approaches such

as support vector machine [20], k-nearest neighbors [11], random forests [24], naive bayes [8] and decision tree [31] have been used to detect and classify known malware. In particular, Nataraj et al. [28] proposed a method for visualizing and classifying malware using image processing methods, which first converts malware binaries to grayscale images. Techniques from computer vision, particularly for image classification can be used to obtain high accuracies.

Researchers have classified malware using CNN models, initially used for image classification [36]. It is obvious that in order to use such an approach, the malware binary must first be converted to an "image". The ANN models used include simple multilayer perceptron, and a mix of GRU-based RNNs and CNNs. Kalash et al. [17] used a CNN model called M-CNN, based on a well-known image classification architecture called VGG-16 [37]. Methods have also replaced the last layer of an artificial neural network with an SVM classifier [30].

In this paper, we compare the performance of several CNN-based models which had achieved state-of-the-art results for malware image classification with the CNN-mixed models used by Agarap and Pepito [2], the CNN models we choose have performed well in the large-scale image classification contest called ILSVRC [34], within the last few years.

The paper is organized in the following way. In the next section, we briefly review related work. Section 3 describes the methodology used to classify malware. Section 4 discusses experimental results. Lastly, Section 5 concludes the paper and discusses plans for future work.

2 Related Work

Below, we discuss research effects that primarily convert malware binaries to images before classifying them. Approaches based on traditional machine learning depend on manual feature extraction. Deep learning can extract useful features automatically by avoiding manual feature extraction.

2.1 Methods Based on Traditional Machine Learning

Grayscale images can be extracted from the raw malware executable files showing features of malware [29] [28] [22]. Such images enable analysis of malware by extracting visual features. Nataraj et al. [28] were the first to explore the use of byte plot visualization as grayscale images for automatic malware classification. They used a malware image dataset consisting of 9,342 malware samples belonging to 25 different classes. They extracted GIST [43] features from the grayscale images and classified them using K-nearest neighbor classification with Euclidean distance as metric. Their approach had high computational overhead. Mirza et al. extracted features from malware files and combined decision trees, support vector machines and boosting to detect malware [27]. Zhang et al. proposed a static analysis technique based on n-grams of opcodes to classify ransomware families [49]. Makandar and Patrot [25] used multi-class support vector machine malware classification with malware input as images. They used wavelet transform to build effective texture based feature vectors from the malware images. This reduced the dimensionality of the feature vector and the time complexity.

2.2 Methods Based on Deep Learning

Several studies on malware classification have been performed using CNN architectures. Cui et al. [6] detected code variants that are malicious after converting to grayscale images and using a simple CNN model. Kalash et al. [17] classified malware images by converting malware files into grayscale images, using two different datasets, Malimg [28] and Microsoft malware [33]. They obtained 98.52% and 99.97% accuracies, respectively. Yue [47] proposed a weighted softmax loss for CNNs for imbalanced malware image classification, and achieved satisfactory classification results. Gilbert. et al. [12] built a model consisting of three convolutional layers with one fully connected layer and tested on two datasets, Microsoft Malware Classification Challenge dataset and Malimg dataset. Seonhee et al. [35] proposed a malware classification model using a CNN that classified malware images. Their experiments were divided into two sets. The first set of experiments classified malware into 9 families and obtained accuracies of 96.2%, 98.4% considering the top-1 and top-2 ranked results. The second set of experiments classified malware into 27 families and obtained 82.9% and 89% top-1 and top-2 accuracies. Tobiyama et al. [42] proposed a malware process detection method by training a recurrent neural network (RNN) to extract features of process behavior, and then training a CNN to classify features extracted by the trained RNN. Vinayakumar et al. proposed a deep learning model based on CNN and LSTM for malware family categorization. Experiments showed an accuracy of 96.3% on the Malimg dataset [46]. Su et al. [38] created one-channel grayscale images from executable binaries in two families, and classified them into their related families using a light-weight convolutional Neural Network. They achieved a accuracies of 94.0% and 81.8% for malware and goodware, respectively.

3 Methodology

In this paper, we use six CNN models for malware classification, considering malware binaries as images.

3.1 Malware Binaries

The malware binaries we use are in Portable Executable (PE) form. Generally, PE files are programs that have file name extensions such as .bin, .dll and .exe. PE files are

usually recognized through their components, which are called .tex, .rdata, .data and .rsrc. The first component, called .text, is the code section, containing the program's instructions. .rdata is the part that contains read only data, and .data is the part that contains data that can be modified, and .rsrc is the final component that stands for resources used by the malware.

Malicious data binaries can be converted 8 bits at a time to pixels in a grayscale image, consisting of textural patterns. In Figure 2, we see the sections of a malware binary showing different textures, when seen as an image [28]. Based on these patterns, we can classify malware. In this paper, we use the Malimg dataset [28] which is a set of grayscale images corresponding to malware binaries saved in .jpg format. Some examples of malware families are shown in Figure 3.

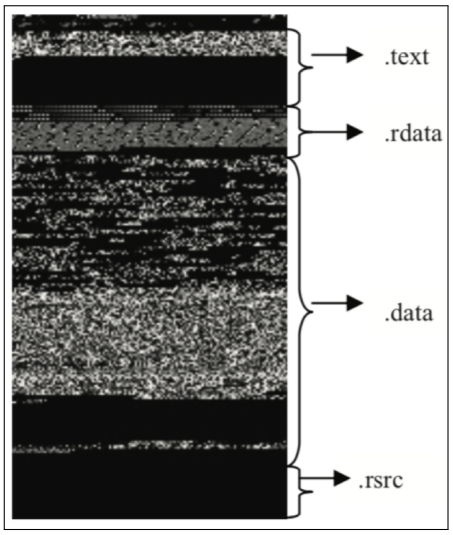


Figure 2: Portable Executable file represented as an image.

3.2 Malware as Image

Researchers and practitioners can understand malware better by visualizing malware binaries as images since, the patterns within such images become clearly visible. Finding patterns within images can be performed well by deep learning [13]. The most important patterns of features in the malware images can be used to identify the malware families also. Images for a specific malware family have similar patterns, allowing a deep learning model to recognize important patterns using automatic extraction of features. In particular, CNN models are good at classifying images because they can extract relevant features within an image by subsampling through convolutions, pooling and other computations. In this case, CNNs look for the most relevant features within an image from a specific malware family for the purpose of classification [6]. Malware binaries can be translated into an images using an algorithm that converts a binary PE file into a sequence of 8 bit vectors or hexadecimal values. An 8 bit

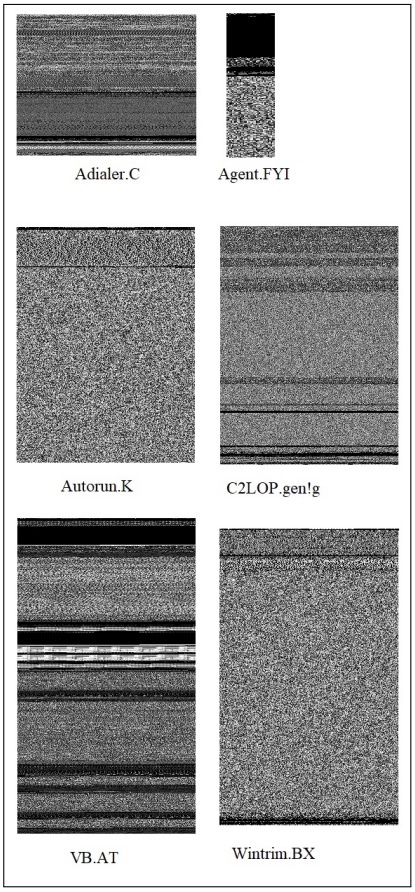


Figure 3: Sample images of malware belonging to different families.

vector can be represented in the range 00000000 (0) to 11111111 (255). Each 8 bit vector represents a number, and can be converted into pixel in a malware image, as shown in Figure 3. Images obtained from different malware families have different characteristics [17].

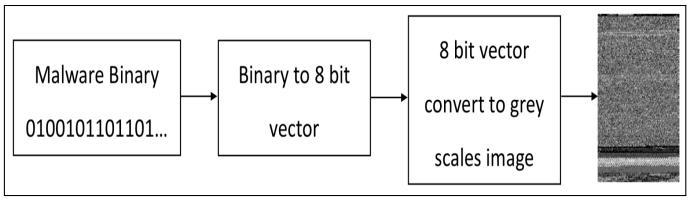


Figure 4: Converting malware binary to an image.

3.3 Problem Statement

The problem that we solve in this paper is classification of malware object code into malware families. We have 9,342 malware samples given in the form of images obtained from their object codes. There are 25 malware families, with the biggest family containing 2,950 samples and the smallest containing 81 samples. We classify these images using deep learning models that have performed well in image classification.

3.4 Motivation and Approach

CNNs have performed well for classification in a variety of domains including object recognition [19], image classification [23], and video classification [18]. CNNs have shown superior performance compared to traditional learning algorithms, especially in tasks such as image classification. Since we represent malware object codes as images, we classify malware based on their corresponding images using CNN models. Malware images are classified into families by extracting patterns within them, because binary image files generated from a malware family are likely to produce similar images. Feature extraction allows image classification models to recognize patterns based on pixel distribution in an image. Before CNNs, features were extracted manually, and it was one of the biggest challenges in image classification. The ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [34] has led to sophisticated CNN-based classification models that have achieved excellent results, demonstrating that the models are likely to perform well in static analysis of malware.

In this paper, we compare the performance of several CNNs-based models for classification of malware binaries that have been converted to images. In particular, we compare the performance of several well-known CNNs-based deep learning models from the ILSVRC competitions and a few additional CNN and CNN-mixed models to classify malware images, that automatically extract features based on the static analysis approach. These models are publicly available.

3.5 CNN Models Used

The experimental work of this paper is to run six deep learning models to classify malware images to detect malware. These models are briefly described below.

3.5.1 VGG16

The first model we use is called VGG-Net16 [37], which was the winner of ILSVRC in 2014. Its contribution was in increasing the depth using 3x3 convolution filters that are small, allowing them to increase the number of layers from 16 to 19. The depth of the representation was very helpful in increasing the accuracy of image classification. On the ImageNet dataset, the VGG model outperformed many complicated models, signifying the importance of the depth.

3.5.2 Inception V3

The Inception V3 model contains 42 layers, and is an improvement over the GoogleNet Inception V1 model that was the winner of ILSVRC in 2015 [39]. The Inception V3 model architecture starts with a 5x Inception module A, 4x Inception module B, 2x Inception module C, and 2x grid size reduction; one of the grid size reductions is done with some modification, and the second one is applied without any modification. An auxiliary classifier is also applied as an extra layer to help improve the results.

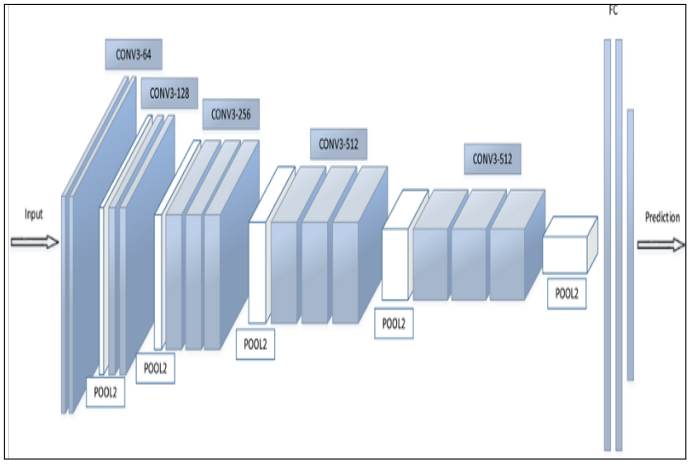


Figure 5: VGG-16 model architecture [3].

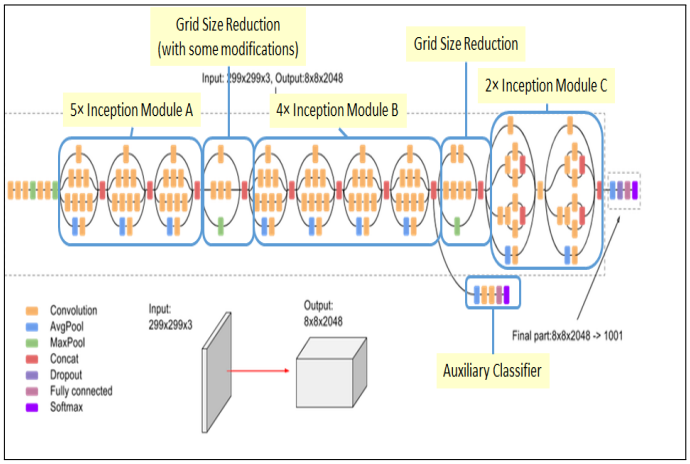


Figure 6: Inception V3 model architecture [45].

3.5.3 ResNet50

The third model we use is called Residual Networks (ResNet50) [15]. ResNet50 was the winner of ILSVRC in 2016. The novel technique that this model introduced provides extra connections between non-contiguous convolutional layers, using shortcut connections. This technique allowed the model to skip through layers to deal with vanishing gradients in order to achieve lower loss and better results. The network had 152 layers, an impressive 8 times deeper than a comparable VGG network. This is an improvement over the VGG16 model with Faster R-CNN, producing an improvement of 28% in accurcy in image classification. The architecture of the original ResNet50 is illustrated in Figure 7.

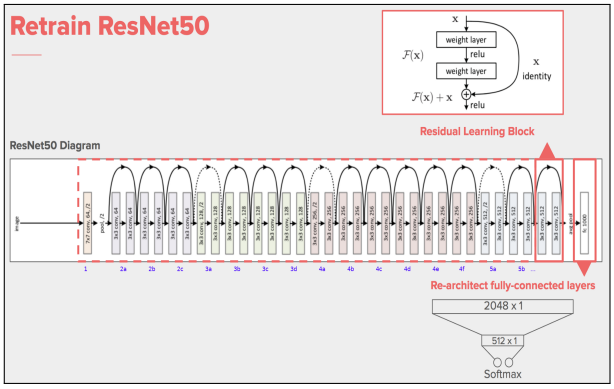


Figure 7: ResNet50 model architecture [5].

3.5.4 CNN-SVM Model

For classification, deep learning models usually use the softmax activation function as the top layer for prediction and minimization of cross-entropy loss. Tang [40] replaced the softmax layer with a linear SVM and applied it on MNIST and CIFAR-10 datasets, and the ICML 2013 Representation Learning Workshop's face expression recognition challenge. The SVM is a linear maximum margin classifier. CNN-SVM allowed for extraction of features for input images with a linear SVM [9]. Agarap and Pepito [2] applied CNN-SVM [40] on Malimg and achieved 77.22% accuracy.

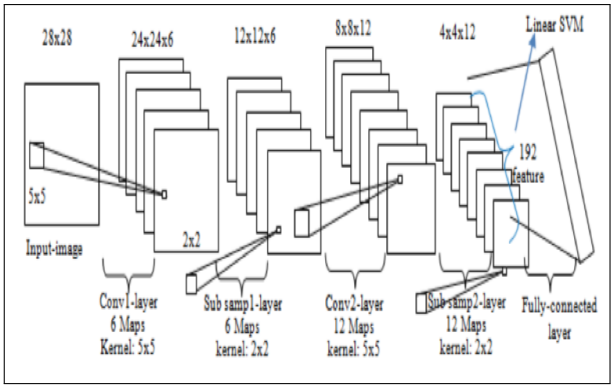


Figure 8: Architecture of CNN-SVM [7].

3.5.5 GRU-SVM Model

Agarap and Pepito [2] modified the architecture of a Gated Recurrent Unit (GRU) RNN by using SVM as its final output layer for use in a binary, non-probabilistic classification task (see Figure 8). They used GRU-SVM on the Malimg dataset and achieved 84.92% accuracy.

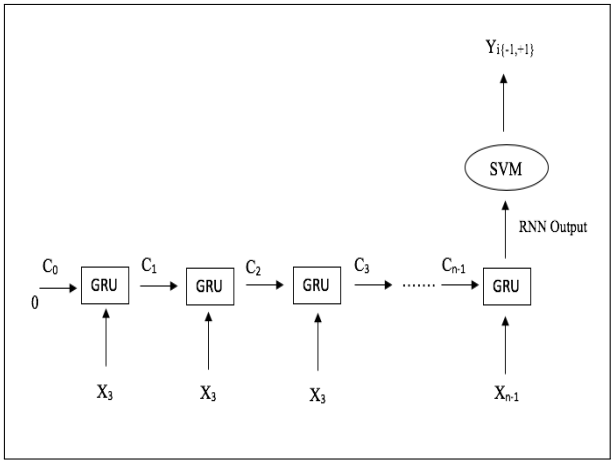


Figure 9: GRU-SVM architecture model, with n GRU cells and SVM for the classification function [1].

3.5.6 MLP-SVM Model

Bellili et al. [4] proposed MLP-SVM for handwritten digit recognition. MLP-SVM is a model that combines both SVM and Multilayer Perceptrons for the classification of binary image. Multilayer Perceptrons are a fully connected network that allows for the inputs to get classified using input features. The MLP-SVM is a hybrid model that run the MLP and SVM classifiers in parallel. The MLP-SVM model was used by Agarap and Pepito [2] on the Malimg dataset with 80.46% accuracy.

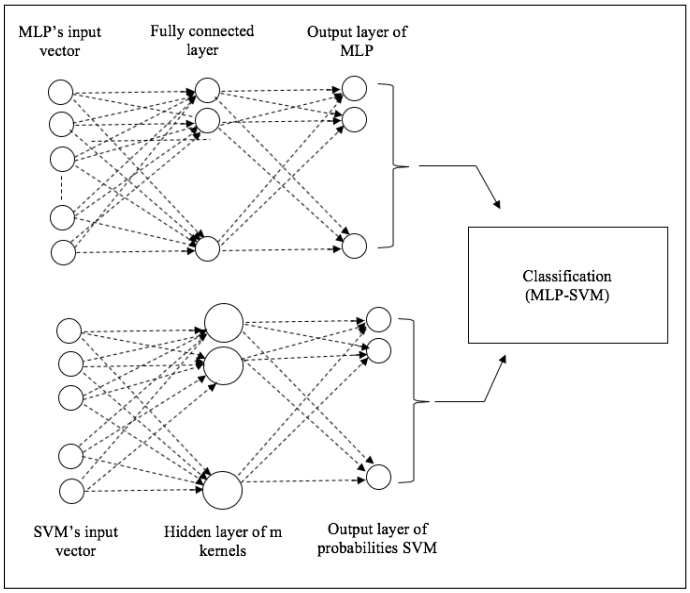


Figure 10: MLP-SVM architecture model [44].

3.6 Dataset

There are a few malware datasets available for academic research. One of the these datasets is Malimg [28]. The

dataset contains 9,342 malware images, classified into 25 malware families. The widths and lengths of the malware images vary. The images have been created from various malware families such as Dialer, Backdoor, Worm, Worm-AutoIT, Trojan, Trojan-Downloader, Rouge and PWS. All malware images are PE files that were first converted to an 8-bit vector binary, and then to images. The malware image sizes were modified, so that they can be input to a CNN model. The family breakdown for the Malimg dataset is shown in Table 1.

Table 1: 25 malware families (classes) and the number of samples in each family.

Malware Family	Samples	Malware kind
Adialer.C	123	Dialer
Agent.FYI	117	Backdoor
Allaple.A	2950	Worm
Allaple.L	1592	Worm
Alueron.gen!J	199	Trojan
Autorun.K	107	Worm AutoIT
C2LOP.gen!g	201	Trojan
C2LOP.p	147	Trojan
Dialplatform.B	178	Dialer
Donoto.A	163	Trojan Downloader
Fakerean	382	Rouge
Instaccess	432	Dialer
Lolyada.AA1	214	PWS
Lolyada.AA2	185	PWS
Lolyada.AA3	124	PWS
Lolyada.AT	160	PWS
Malex.gen!J	137	Trojan
Obfuscator.AD	143	Trojan Downloader
RBot!gen	159	Backdoor
Skintrim.N	81	Trojan
Swizzor.gen!E	129	Trojan Downloader
Swizzor.gen!I	133	Trojan Downloader
VB.AT	409	Worm
Wintrim.BX	98	Trojan Downloader
Yuner.A	801	Worm

4 Experimental Results

All experiments in this study were conducted on NVIDIA GeForce GTX 1080 Ti GPU. As stated, we ran six models on the Malimg dataset: Inception V3, VGG16-Net, ResNet50, CNN-SVM, MLP-SVM and GRU-SVM. Since the Malimg dataset is not similar to the ImageNet dataset, we could not directly use grayscale images with VGG16 and ResNet50 because the input layers require the shape of (3, 224, 224). The 3 is represents Red, Green and Blue (RGB) channels of the image, whereas the grayscale images require (1, 224, 224). VGG16 and ResNet50 showed low performance compared to the other models, since both of these models architectures were designed to recognize colored images that requires RGB format. Therefore, both give low accuracies when tested on the grayscale images. The results for malware prediction using all these models are shown in Table 2 and Figure 11. The Inception V3 model had a significantly higher accuracy at 99.24%. Table 4 shows the best predicted accuracies of the six models when run 10 times. CNN-SVM, GRU-SVM, and MLP-SVM performed well but VGG16 and ResNet50 performed poorly compared to the Inception V3 model. We provide the results of testing the dataset with several traditional models as well as other deep learning models in Table 4.

5 Conclusions and Future Work

These days many antivirus programs rely on deep learning techniques to protect devices from malware. Deep learning architectures have achieved good performance in detecting malware when used with Windows PE binaries. We have presented the performance comparison among six classifiers on a malware image dataset created from PE files. We used the models from the ImageNet Large-Scale Visual Recognition Challenge and three other CNN models to classify grayscale malware images. We successfully trained the six models on the Malimg dataset, and the results indicate that the Inception-V3 model outperforms all compared work. To the best of our knowledge, it is the state-of-the-art of performance in classification on grayscale malware images.

Future work will be focused on conducting results using additional models from leaderboards of image classification competitions. We also want to convert malware images into color RGB images before classification.

References

[1] A. Abien, "A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data," in *The 10th International Conference on Machine Learning and Computing*, pp. 26–30, 2018.

[2] A. F. Agarap and F. J. H. Pepito, "Towards building an intelligent anti-malware system: A deep learning approach using support vector machine (SVM) for malware classification," *ArXiv*, 2017. (`arXiv:1801.00318`)

[3] S. Bansal, *CNN Architectures : VGG, ResNet, Inception + TL*, 2018. (`https://www.kaggle.com/shivamb/cnn-architectures-vgg-resnet-inception-tl`).

[4] A. Bellili, M. Gilloux, and P. Gallinari, "An hybrid MLP-SVM handwritten digit recognizer," in *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pp. 28–32, 2001.

[5] A. Ciurana, *How to split resnet50 model from top as well as from bottom?*, 2019. (`https://stackoverflow.`

Table 2: Prediction accuracies of the six tested models.

Family	CNN-SVM	GRU-SVM	MLP-SVM	Inception V3	ResNet 50	VGG16
	Prediction Accuracy					
Adialer.C	**99.80%**	99.15%	99.51%	99.40%	23.18%	13.62%
Agent.FYI	95.12%	95.86%	94.87%	**99.50%**	25.41%	14.81%
Allaple.A	94.98%	97.71%	94.32%	**99.72%**	26.94%	14.47%
Allaple.L	99.10%	95.35%	95.48%	**99.73%**	21.52%	15.53%
Alueron.gen!J	96.42%	97.57%	93.20%	**99.48%**	21.37%	15.93%
Autorun.K	92.99%	93.38%	96.68%	**99.06%**	23.27%	14.38%
C2LOP.gen!g	94.75%	93.70%	94.49%	**98.42%**	28.87%	13.78%
C2LOP.P	97.11%	93.45%	95.43%	**99.67%**	27.48%	14.96%
Dialplatform.B.	95.34%	94.85%	96.17%	**99.86%**	23.84%	14.78%
Dontovo.A	97.53%	89.81%	93.44%	**98.25%**	29.76%	15.03%
Fakerean	98.46%	92.11%	93.11%	**98.91%**	26.29%	12.45%
Instantaccess.	93.17%	96.75%	96.63%	**98.24%**	30.15%	13.11%
Lolyda.AA1	91.30%	94.09%	93.97%	**99.40%**	23.79%	14.26%
Lolyda.AA2	89.10%	94.36%	91.64%	**99.34%**	28.32%	13.80%
Lolyda.AA3	87.44%	90.61%	94.13%	**97.39%**	29.59%	13.85%
Lolyda.AT	81.31%	92.51%	90.28%	**99.86%**	31.67%	13.99%
Malex.gen!J	88.79%	94.99%	94.61%	**99.31%**	25.39%	15.22%
Obfuscator.AD.	86.57%	94.76%	96.74%	**99.50%**	21.84%	12.64%
Rbot!gen.	87.60%	93.39%	97.19%	**98.81%**	32.49%	14.45%
Skintrim.N	96.16%	84.10%	87.21%	**99.55%**	34.81%	15.84%
Swizzor.gen!E.	82.45%	96.72%	98.54%	**99.57%**	17.22%	15.30%
Swizzor.gen!I	97.57%	98.14%	96.80%	**99.29%**	33.57%	14.55%
VB.AT	99.36%	98.72%	98.77%	**99.34%**	31.68%	13.92%
Wintrim.BX	99.78%	97.71%	**99.92%**	99.88%	31.71%	15.65%
Yuner.A	88.26%	84.44%	80.64%	**99.79%**	16.38%	11.54%

Table 3: Accuracy averages of the six tested models.

Models	CNN-SVM	GRU-SVM	MLP-SVM	Inception V3	ResNet 50	VGG16
	Average of prediction accuracy					
	93.22%	94.17%	94.55%	99.25%	26.66%	14.31%

Table 4: Comparison of malware detection models, including models we tested.

Model	VGG16	ResNet50	MLP-SVM [2]	CNN-SVM [2]	GRU-SVM [2]	Random Forest [10]	MLP-SVM	GRU-SVM	CNN [16]	M-CNN [17]	CNN-SVM	Inception V3
Accuracy	15.92	35.10%	80.46%	77.22%	84.92%	95.26%	97.25%	97.43%	98.00%	98.52%	99.11%	**99.24%**

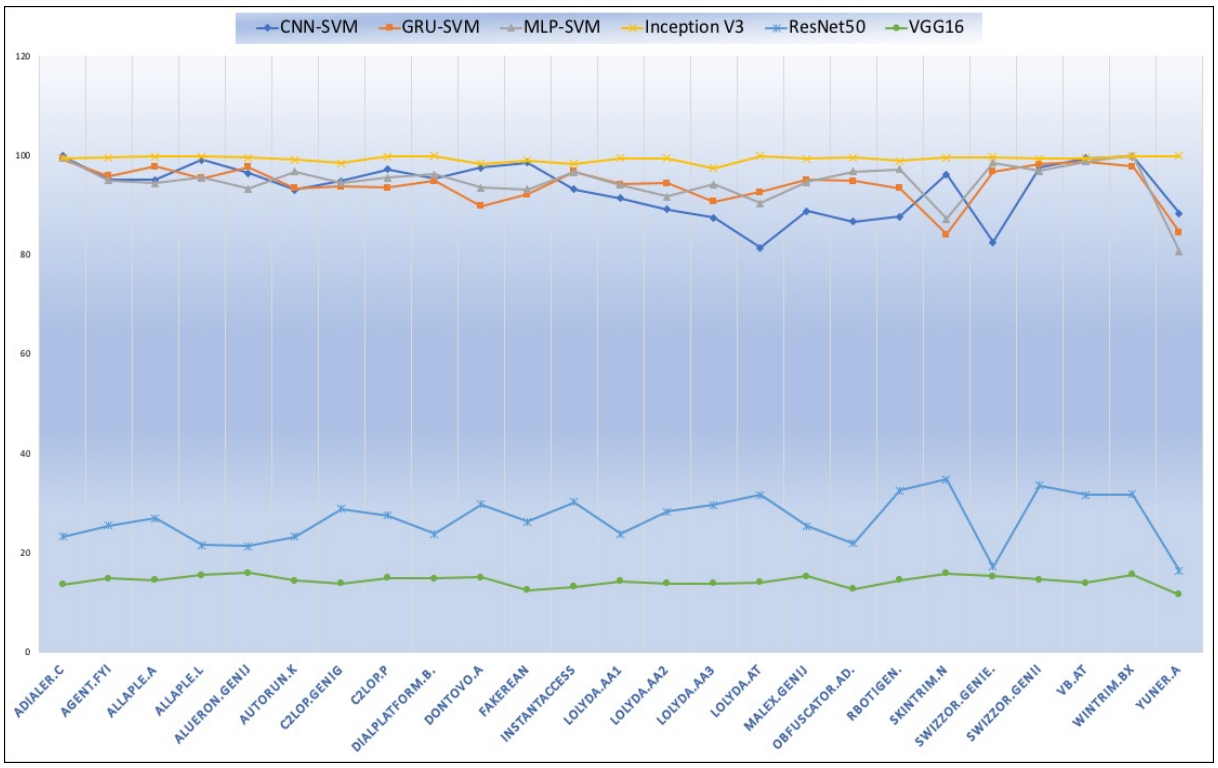


Figure 11: Prediction accuracy of six models

com/questions/54207410/how-to-split-resnet50-model-from-top-as-well-as-from-bottom).

[6] Z. Cui, F. Xue, X. Cai, Y. Cao, G. G. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[7] Darmatasia and M. I. Fanany, "Handwriting recognition on form document using convolutional neural network and support vector machines (CNN-SVM)," *The 5th International Conference on Information and Communication Technology*, pp. 1–6, 2017.

[8] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, no. 2-3, pp. 103–130, 1997.

[9] M. I. Fanany, "Handwriting recognition on form document using convolutional neural network and support vector machines (CNN-SVM)," in *The 5th International Conference on Information and Communication Technology*, pp. 1–6, 2017.

[10] F. C. C. Garcia, I. I. Muga, and P. Felix, "Random forest for malware classification," *Cryptography and Security*, 2016. (arXiv:1609.07770)

[11] F. Gianfelici, "Nearest-neighbor methods in learning and vision," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 377–377, 2008.

[12] D. Gibert, C. Mateu, J. Planes, and R. Vicens, "Using convolutional neural networks for classification of malware represented as images," *Journal of Computer Virology and Hacking Techniques*, vol. 15, no. 1, pp. 15–28, 2019.

[13] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, and J. Cai, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, no. 354–377, 2018.

[14] C. Guarnieri, M. Schloesser, J. Bremer, and A. Tanasi, "Cuckoo Sandbox open source automated malware analysis," *Black Hat USA*, 2013. (https://media.blackhat.com/us-13/US-13-Bremer-Mo-Malware-Mo-Problems-Cuckoo-Sandbox-WP.pdf)

[15] 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.

[16] E. K. Kabanga and C. H. Kim, "Malware images classification using convolutional neural network," *Journal of Computer and Communications*, vol. 6, no. 01, pp. 153, 2017.

[17] M. Kalash, M. Rochan, N. Mohammed, N. Bruce, Y. Wang, and F. Iqbal, "Malware classification with deep convolutional neural networks," in *The 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS'18)*, pp. 1–5, 2018.

[18] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

[19] K. Kavukcuoglu, P. Sermanet, Y. L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun, "Learning convolutional feature hierarchies for visual recognition," in *Advances in Neural Information Processing Systems*, pp. 1090–1098, 2010.

[20] S. S. Keerthi and E. G. Gilbert, "Convergence of a generalized SMO algorithm for SVM classifier design," *Machine Learning*, vol. 46, no. 1-3, pp. 351–360, 2002.

[21] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Australasian Joint Conference on Artificial Intelligence*, pp. 137–149, 2016.

[22] K. Kosmidis and C. Kalloniatis, "Machine learning and images for malware detection and classification," in *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, pp. 1–6, 2017.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[24] A. Liaw and M. Wiener, "Classification and regression by random forest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[25] A. Makandar and A. Patrot, "Malware class recognition using image processing techniques," in *International Conference on Data Management, Analytics and Innovation*, pp. 76–80, 2017.

[26] A. Marx, G. Habicht, and M. Morgenstern, *Malware Statistics and Trends Report*, the AV-TEST Institute, Apr. 2019. (`http://www.av-test.org/en/statistics/malware`)

[27] Q. K. A. Mirza, I. Awan, and M. Younas, "Cloudintell: An intelligent malware detection system," *Future Generation Computer Systems*, vol. 86, pp. 1042–1053, 2018.

[28] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, pp. 4, 2011.

[29] L. Nataraj, V. Yegneswaran, P. Porras, and J. Zhang, "A comparative assessment of malware classification using binary texture analysis and dynamic analysis," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, pp. 21–30, 2011.

[30] X. X. Niu and C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, 2012.

[31] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[32] Robinson and Cole, *Data Privacy and Cybersecurity for Tax Professionals*, IRS nationwide tax forum, 2019. (`https://www.irs.gov/pub/irs-utl/2019ntf-11.pdf`)

[33] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, *Microsoft Malware Classification Challenge (BIG 2015)*, 2018. (`https://www.kaggle.com/c/malware-classification`)

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[35] S. Seok and H. Kim, "Visualized malware classification based on convolutional neural network," *Journal of The Korea Institute of Information Security & Cryptology*, vol. 26, no. 1, pp. 197–208, 2016.

[36] M. Sewak, S. K. Sahay, and H. Rathore, "Comparison of deep learning and the classical machine learning algorithm for the malware detection," in *The 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD'18)*, pp. 293–296, 2018.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Computational and Biological Learning Society Conference at ICLR*, pp. 1–14, 2015.

[38] J. Su, V. D. Vasconcellos, S. Prasad, S. Daniele, Y. Feng, and K. Sakurai, "Lightweight classification of IoT malware based on image recognition," in *IEEE the 42nd Annual Computer Software and Applications Conference*, vol. 2, pp. 664–669, 2018.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[40] Y. Tang, "Deep learning using linear support vector machines," in *Workshop on Challenges in Representation Learning ICML*, 2013. (`arXiv:1306.0239`)

[41] A. Tayal, N. Mishra, and S. Sharma, "Active monitoring and postmortem forensic analysis of network threats: A survey," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 49–59, 2017.

[42] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with deep neural network using process behavior," in *IEEE the 40th Annual Computer Software and Applications Conference*, vol. 2, pp. 577–582, 2016.

[43] A. Torralba, K. P. Murphy, W. T. Freeman, M. A. Rubin, *et al.*, "Context-based vision system for place and object recognition," *International conference of Computer Vision*, vol. 3, pp. 153–167, 2003.

[44] V. Tra, S. Khan, and J. Kim, "Diagnosis of bearing defects under variable speed conditions using energy distribution maps of acoustic emission spectra and convolutional neural networks," *The Journal of the Acoustical Society of America*, vol. 144, no. 4, 2018.

[45] S. H. Tsang, *Review: Inception-v3 — 1st Runner Up (Image Classification) in ILSVRC 2015*, 2018. (`https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c`).

[46] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust intelligent malware detection using deep learning," *IEEE Access*, vol. 7, pp. 46717–46738, 2019.
[47] S. Yue, "Imbalanced malware images classification: A CNN based approach," *Computer Vision and Pattern Recognition*, 2017. (arXiv:1708.08042)
[48] M. Zareapoor, P. Shamsolmoali, and M. A. Alam, "Establishing safe cloud: Ensuring data security and performance evaluation," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 88–99, 2014.
[49] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on N-gram of opcodes," *Future Generation Computer Systems*, vol. 90, pp. 211–221, 2019.

Biography

Ahmed Bensaoud received the B.S. degree from the Benghazi University, Libya, and M.S. from Colorado State University, Fort Collins, Colorado. Currently he is a Ph.D. student at the University of Colorado Colorado Springs. His research interests include malware detection and malware classification.

Nawaf Abudawaood graduated from the University of Colorado at Colorado Springs with a Masters in Engineering in Information Assurance. He received his Bachelors degree from the Old Dominion University Norfolk, Virginia, in Information Systems and Technology. He currently works for The Exchange Hub as a Cyber Security Engineer.

Jugal Kalita received Ph.D. from the University of Pennsylvania, Philadelphia. He is a Professor of Computer Science at the University of Colorado, Colorado Springs. His research interests are in machine learning and natural language processing. He has published over 250 papers in international journals and referred conference proceedings and has written four books.

Analysis of One Fully Homomorphic Encryption Scheme in Client-Server Computing Scenario

Yang Li and Lihua Liu
(Corresponding author: Yang Li)

Department of Mathematics, Shanghai Maritime University,
Haigang Ave 1550, Shanghai, 201306, China
(Email: liyangsmu@126.com, liulh@shmtu.edu.cn)
(Received May 20, 2019; Revised and Accepted Dec. 6, 2019; First Online Feb. 1, 2020)

Abstract

Fully homomorphic encryption (FHE) is a useful primitive which allows anyone to perform arbitrary operations on encrypted data without decryption key, but any arithmetic (such as addition and multiplication) must be constrained to the underlying domain (finite fields or rings). In this paper, we revisit Dasgupta-Pal FHE scheme, and discuss its applications in client-server scenario. We find its calculation process cannot be practically completed due to the flawed relationship between the key size and the length of the plaintext. We would like to stress that the main purpose of cryptography using modular arithmetic is to obscure and dissipate redundancies in plaintext, not to perform numerical calculations. We think FHE could be of little significance in client-server scenario.

Keywords: Client-Server Computing; Fully Homomorphic Encryption; Polynomial Rings; Symmetric Encryption

1 Introduction

In recent years, fully homomorphic encryption (FHE) has flourished in various fields such as cloud computing, economic, searchable encryption, spam filtering, watermarking, e-voting, and medical services [16–18, 22, 29], which can be used to protect user privacy. Homomorphic encryption (HE) supporting either addition or multiplication (but not both), has some limitations for various scenarios.

In 1978, Rivest *et al.* [25] investigated the problem of privacy homomorphisms. Paillier's encryption [21] was a popular quasi-homomorphic encryption. In 2009, Gentry [10] presented a fully homomorphic encryption over ideal lattices which can allow anyone to evaluate the circuit on the encrypted data without the decryption key. But it is very slow because its key size is too large. In 2010, Gentry *et al.* [13] proposed a simple BGN-type cryptosystem from LWE. Shortly afterwards, Gentry and Halevi [11] presented a fully homomorphic encryption free of using depth-3 arithmetic circuits. In 2010, Dijk *et al.* [9] constructed an integer-based FHE scheme using modular operations. At Crypto'11, Coron *et al.* [7] provided a FHE scheme over integers with shorter public key. In 2013, Gupta and Sharma [14] presented a FHE scheme using a symmetric key with a smaller size. The works [3, 12, 28] tried to optimize ciphertext length and to improve efficiencies of some FHE schemes. In 2015, Nuida and Kurosawa constructed a fully homomorphic encryption scheme over integers with the message space $\mathbb{Z}_Q$ for any prime Q. The later works [2, 5, 6, 20, 27] studied the possible applications and optimizations of FHE. In 2019, Salavi *et al.* [26] considered the combinations of some traditional and modern cryptographic techniques for designing FHE schemes.

Cloud computing is a promising innovation for storing large amount of data. Sensitive data stored on cloud platforms are vulnerable to attacks by hackers and unauthorized parties. The works [19, 23, 24] discussed how to protect the security of cloud computing. In 2016, Jeng *et al.* [15] proposed a method to resist attacks against cloud storage services. Dasgupta and Pal [8] designed a symmetric FHE scheme based on polynomial rings. In 2018, Aganya and Sharma [1] improved the FHE scheme. Cao *et al.* [4] pointed out that some typical FHE schemes were not suitable for client-server or cloud computing scenarios.

In this paper, we revisit the Dasgupta-Pal FHE scheme and analyze its applications in client-server scenario. We would like to point out that the scheme fails to draw a clear line between numerical operations and modular operations. The relationship between the key size and the length of the plaintext is also confused. We want to stress that any computations operated on encrypted data should be constrained to the underlying domain. Otherwise, the related computations will generate some wrong outputs.

2 Dasgupta-Pal FHE Scheme

2.1 Description

The scheme can be described as follows.

KeyGen. For a security parameter l, generate secret key S_k. Pick a prime number of l bits and an even integer z of length γ. Set $R_k = z \cdot S_k$.

Enc. Pick n-degree polynomial $y(x), d(x)$ such that $m_p(x) \equiv y(x) \bmod S_k$, coefficients of $d(x)$ are integers of length l^a. Compute $c(x) = y(x) + S_k \cdot d(x)$.

Dec. Compute $c(x) \bmod S_k \bmod 2 = m_p(x)$.

Refresh. Compute $c'(x) = c(x) \bmod R_k$.

Clearly, we have

$$c_1(x) + c_2(x) \bmod S_k \bmod 2 = m_{p1}(x) + m_{p2}(x),$$
$$c_1(x) \cdot c_2(x) \bmod S_k \bmod 2 = m_{p1}(x) \cdot m_{p2}(x).$$

2.2 An Example in Client-Server Scenario

Suppose that $S_k = 13$. There are two numbers $m_1 = 69$ and $m_2 = 57$, a client asks a server to check if $(c_1 + c_2)(x) \bmod S_k \bmod 2$ is equal to $(m_{p1} + m_{p2})(x)$. Now, the scheme will encrypt m_1 and m_2 as follows.

$$\begin{aligned} m_1 &= 69 \to (1000101)_2 \to x^6 + x^2 + 1, y_1(x) \\ &\equiv m_1(x) \bmod 13 = 27x^6 + 14x^2 + 14. \end{aligned}$$

If pick $d(x) = 2650x^6 + 995x^5 + 259x^2 + 100$. Then

$$\begin{aligned} c_1(x) &= y_1(x) + S_k \cdot d_1(x) \\ &= 34477x^6 + 12935x^5 + 3381x^2 + 1314. \end{aligned}$$

$$\begin{aligned} m_{p1}(x) &= c_1(x) \bmod S_k \bmod 2 \\ &= (34477x^6 + 12935x^5 \\ &\quad +3381x^2 + 1314) \bmod 13 \bmod 2 \\ &= x^6 + x^2 + 1. \end{aligned}$$

$$\begin{aligned} m_2 &= 57 \to (111001)_2 \\ &\to x^5 + x^4 + x^3 + 1 \in R[x], \\ y_2(x) &\equiv m_2(x) \bmod 13 = 14x^5 + 27x^4 + 40x^3 + 14. \end{aligned}$$

If pick $d(x)$ $d_2(x) = 119x^5 + 224x^4 + 17x^3 + 2249x^2 + 36$. Then

$$\begin{aligned} c_2(x) &= y_2(x) + S_k \cdot d(x) \\ &= 1561x^5 + 2939x^4 + 261x^3 + 29237x^2 + 482. \end{aligned}$$

$$\begin{aligned} m_{p2}(x) &= c_2(x) \bmod S_k \bmod 2 \\ &= (1561x^5 + 2939x^4 + 261x^3 \\ &\quad +29237x^2 + 482) \bmod 13 \bmod 2 \\ &= x^5 + x^4 + x^3 + 1. \end{aligned}$$

2.3 Analysis

Suppose a, b are in the domain of one FHE encryption algorithm $E(\cdot)$, and $D(\cdot)$ is the corresponding decryption algorithm. Hence,

$$D(E(a) + E(b)) = D(E(a+b)) = (a+b) \bmod p$$
$$D(E(a) \cdot E(b)) = D(E(a \cdot b)) = (a \cdot b) \bmod p.$$

where p is the associated modular. Generally,

$$a + b \neq (a+b) \bmod p, \quad a \cdot b \neq (a \cdot b) \bmod p$$
$$a < b \not\Longrightarrow E(a) < E(b),$$
$$a \bmod q \bmod p \not\Longrightarrow a \bmod p \bmod q.$$

Any modular computations are constrained to the underlying domain, such as finite fields and rings. The above scheme uses only elementary modular operations on polynomial rings. But as we see that modular arithmetic is mainly used to obscure and dissipate redundancies in plaintext, not for common numerical calculations.

In the above example, the server who is not knowing the secret key will generate the following results for addition and multiplication.

$$\begin{aligned} & c_1(x) + c_2(x) \bmod 13 \bmod 2 \\ = \; & 34477x^6 + 14496x^5 + 2939x^4 + 261x^3 \\ & +32618x^2 + 1796 \bmod 13 \bmod 2 \\ = \; & x^6 + x^5 + x^4 + x^3 + x^2 \to (1111100)_2 = 124. \end{aligned}$$

But $m_1 + m_2 = 69 + 57 = 126 \neq 124$.

$$\begin{aligned} & c_1(x) \cdot c_2(x) \bmod S_k \bmod 2 \\ = \; & 53818597x^{11} + 121519438x^{10} + 47014462x^9 \\ & +1011380084x^8 + 383458336x^7 + 9936759x^6 \\ & +9168265x^5 + 98850297x^4 + 342954x^3 + 40047060x^2 \\ & +633348 \bmod 13 \bmod 2 \\ = \; & x^{11} + x^{10} + x^9 + x^7 + x^4 \\ & +x^3 + x^2 + 1 \to (111010011101)_2 \\ = \; & 3741. \end{aligned}$$

But But $m_1 \times m_2 = 69 \times 57 = 3933 \neq 3741$.

That is to say, the server will generate wrong outputs even for the two simple operations. In short, the scheme is not suitable for client-server scenario, because the overflow errors. Besides, it can not deal with any rounding errors in common numerical computations.

3 Aganya-Sharma FHE Scheme

In 2018, Aganya and Sharma [1] tried to improve Dasgupta-Pal FHE scheme. For readers' convenience, we now describe it as follows.

3.1 Description

The message space is $\{0,1\}^p$, where p is called batch size. l is a security parameter. Divide m into n substrings

$m^{(1)}, \ldots, m^{(n)}$, each of p bits. If the length of m is not an integer multiple of p, then add some padding bits 0s at its right end until it becomes a p bits string. For encryption purposes, each p bits string $m^{(i)}$ is converted into a decimal integer $m^{(i)}$.

KeyGen. Generate S_k, a prime number of l bits. Choose an even integer z of length γ, where $\gamma = \log_2 l$. Then set $R_k = z \cdot S_k$.

Enc. Choose a polynomial $y(x)$ of degree n such that $m_p(x) \equiv y(x) \bmod S_k$. Pick a polynomial $d(x)$ of degree n, with coefficients are integer of length l^a. Compute $c(x) = y(x) + S_k \cdot d(x)$.

Dec. Compute $c(x) \bmod S_k \bmod 2^p = m_p(x)$.

Refresh. Compute $c'(x) = c(x) \bmod R_k$.

3.2 Example

Suppose that $m_1 = 69$, $m_2 = 57$, $a = 5$. Set $S_k = 13$ be the secret key. $m_1 = 69 = (1000101)_2$, $m_2 = 57 = (111001)_2$. Pad $m^{(n)}$ with some 0s to ensure its length is an integer multiple of 3. Then $p = 3$ and the chunks are 001, 000, 101. In client-server scenario, its encrypting process is described as follows Table 1.

Table 1: m_1 and m_2 chunks

Encrypt 1:	$y_1^1 = 2453972,\ d_1^1 = 995,\ c_1^1 = 2466907$
Encrypt 0:	$y_1^2 = 445042,\ d_1^2 = 874,\ c_1^2 = 456404$
Encrypt 5:	$y_1^3 = 300708,\ d_1^3 = 742,\ c_1^3 = 310354$
Encrypt 0:	$y_2^1 = 445042,\ d_2^1 = 874,\ c_2^1 = 456404$
Encrypt 7:	$y_2^2 = 834750,\ d_2^2 = 731,\ c_2^2 = 844253$
Encrypt 1:	$y_2^3 = 2453972,\ d_2^3 = 995,\ c_2^3 = 2466907$

We shall obtain the coefficients 2923311, 1300657, 2777261. Its decrypting process will return the sum $(1111110)_2 = 126$.

3.3 Analysis

In the above example, the server who is not knowing the secret key will return wrong output for other texts. In fact, if choose $m_3 = 123 = (1111011)_2$ and $m_4 = 181 = (10110101)_2$. Adding 0s to the left ends of $m^{(3)}$ and $m^{(4)}$, we shall obtain the following two ciphertexts:

$$\underbrace{2466907, 844253, 315955}_{c_3} \quad \text{and} \quad \underbrace{458057, 453251, 310354}_{c_4}$$

For the function $f(x, y) = x + y$, the server evaluates it with the ciphertexts. It then returns values: 2924964, 1297504, 626309. Therefore, the decrypting process will return values:

$$\begin{aligned} 2924964 &\bmod S_k \bmod 2^p = 3, \\ 1297504 &\bmod S_k \bmod 2^p = 0, \\ 626309 &\bmod S_k \bmod 2^p = 0. \end{aligned}$$

Clearly, $(11000000)_2 = 192$, which is not the wanted value, 304. That is to say, the scheme has also confused the common numerical operations with the underlying modular operations. Namely, the improvement is also unsuitable for client-server scenario in cloud computing.

3.4 Further Discussions

What computations do you want to outsource privately? Backup your phone's contact directory to the cloud? Ask the cloud to solve a mathematic problem in your homework? Do a private web search? $\cdots$. It seems obvious that the daily computational tasks are rarely constrained to some prescribed modulus. Moreover, the client-server computing model can not deal with relational expressions which are defined over plain data, not over encrypted data. This is because

$$a < b \not\Longrightarrow E(a) < E(b), \quad E(a) < E(b) \not\Longrightarrow a < b.$$

In view of this drawback of FHE and the flaws discussed above, we think FHE seems inappropriate for the scenario of cloud computing.

4 Conclusion

In this note, we analyze two FHE schemes over polynomial rings in client-server scenario. We find the problem that what computations are worth delegating privately by individuals and companies to untrusted devices or servers remains untouched. We think the cloud computing community has not yet found a good for-profit model convincing individuals to pay for this or that computational service.

Acknowledgements

We thank the National Natural Science Foundation of China (Project 61411146001). The authors are very grateful to the reviewers for their valuable suggestions.

References

[1] K. Aganya and I. Sharma, "Symmetric fully homomorphic encryption scheme with polynomials operations," in *Proceedings of IEEE Second International Conference on Electronics, Communication and Aerospace Technology (ICECA'18)*, pp. 1954–1957, Mar. 2018.

[2] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[3] Z. Brakerskim, C. Gentry, and V. Vaikuntanathan, "(Leveled) Fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory*, vol. 6, no. 3, pp. 1–36, 2014.

[4] Z. J. Cao, L. H. Liu, and Y. Li, "Ruminations on fully homomorphic encryption in client-server computing scenario," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 32–39, 2018.

[5] G. Castagnos and F. Laguillaumie, "Linearly homomorphic encryption from DDH," in *Proceedings of Topics in Cryptology, The Cryptographer's Track at the RSA Conference (CT-RSA'15)*, pp. 487–505, Apr. 2015.

[6] J. H. Cheon and J. Kim, "A hybrid scheme of public-key encryption and somewhat homomorphic encryption," *IEEE Transaction on Information Forensics and Security*, vol. 10, no. 5, pp. 1052–1063, 2015.

[7] J. Coron and *et al.*, "Fully homomorphic encryption over the integers with shorter public keys," in *Proceedings of 31st Annual Cryptology Conference, Advances in Cryptology (CRYPTO'11)*, pp. 487–504, Aug. 2011.

[8] S. Dasgupta and S. K. Pal, "Design of a polynomial ring based symmetric homomorphic encryption scheme," *Perspectives in Science*, vol. 8, no. C, pp. 692–695, 2016.

[9] M. Dijk and *et al.*, "Fully homomorphic encryption over the integers," in *Proceedings of 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT'10)*, pp. 24–43, June 2010.

[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] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," in *Proceedings of IEEE Annual Symposium on Foundations of Computer Science (FOCS'11)*, pp. 107–116, Palm Springs, Oct. 2011.

[12] C. Gentry and S. Halevi, "Implementing gentry's fully homomorphic encryption scheme," in *Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT'11)*, pp. 129–148, May 2011.

[13] C. Gentry, S. Halevi, and V. Vaikuntanathan, "A simple bgn-type cryptosystem from LWE," in *Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT'10)*, pp. 506–522, May 2010.

[14] C. P. Gupta and I. Sharma, "A fully homomorphic encryption scheme with symmetric keys with application to private data processing in clouds," in *Proceedings of IEEE Fourth International Conference on the Network of the Future (NOF'13)*, pp. 23–25, Oct. 2013.

[15] F. G. Jeng and *et al.*, "On the security of privacy-preserving keyword searching for cloud storage services," *International Journal of Network Security*, vol. 18, no. 3, pp. 597–600, 2016.

[16] J. Kim, S. Kim, and J. H. Seo, "A new scale-invariant homomorphic encryption scheme," *Information Sciences*, vol. 422, pp. 177-187, 2017.

[17] C. L. Liu and C. W. Hsu, "Comment on 'improved secure RSA cryptosystem (ISRSAC) for data confidentiality in cloud'," *International Journal of Network Security*, vol. 21, no. 4, pp. 709–712, 2019.

[18] L. H. Liu and Z. J. Cao, "Analysis of two confidentiality-preserving image search schemes based on additive homomorphic encryption," *International Journal of Electronics and Information Engineering*, vol. 5, no. 1, pp. 1–5, 2016.

[19] L. H. Liu, Z. J. Cao, and C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, pp. 29–35, 2018.

[20] K. Nuida and K. Kurosawa, "(Batch) Fully homomorphic encryption over integers for non-binary message spaces," in *Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT'15)*, pp. 537–555, Apr. 2015.

[21] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT'99)*, pp. 223–238, May 1999.

[22] M. M. Potey, C. A. Dhote, and D. H. Sharma, "Homomorphic encryption for security of cloud data," *Procedia Computer Science*, no. 79, pp. 175–181, 2016.

[23] V. S. Rao and N. Satyanarayana, "On multi-user based efficient computation outsourcing scheme and its application to cloud," *International Journal of Network Security*, vol. 21, no. 2, pp. 303–311, 2019.

[24] S. Rezaei, M. A. Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115–131, 2018.

[25] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation, Academia Press*, pp. 169–180, 1978. (`http://luca-giuzzi.unibs.it/corsi/Support/papers-cryptography/RAD78.pdf`)

[26] R. R. Salavi, M. M. Math, and U. P. Kulkarni, "A survey of various cryptographic techniques: From traditional cryptography to fully homomorphic encryption," in *Proceedings of Innovations in Computer Science and Engineering (ICICSE'19)*, pp. 295–305, Aug. 2019.

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

[28] B. C. Wang, Y. Zhan, and Z. L. Zhang, "Cryptanalysis of a symmetric fully homomorphic encryption scheme," *IEEE Transaction on Information and Forensics Security*, vol. 13, pp. 1460–1467, 2018.

[29] C. H. Wei, M. S. Hwang, and A. Y. Chin, "A secure privacy and authentication protocol for passive rfid tags," *International Journal of Mobile Communications*, vol. 15, no. 3, pp. 266–277, 2017.

Biography

Yang Li is currently pursuing his M.S. degree from Department of Mathematics, Shanghai Maritime university. His research interests include combinatorics and cryptography.

Lihua Liu is an associate professor with Department of Mathematics at Shanghai Maritime University. She received her Ph.D. degree in applied mathematics from Shanghai Jiao Tong University. Her research interests include combinatorics, cryptography and information security.

Attack-Defense Game Model: Research on Dynamic Defense Mechanism of Network Security

Xuhua Zhao

(Corresponding author: Xuhua Zhao)

Center of Information and Technology, Zhuhai City Polytechnic
Jiner Road, Xihu Urban Community, Jinwan District, Zhuhai, Guangdong 519090, China
(Email: xh_zhaoxh@126.com)

(Received Apr. 13. 2020; revised and accepted Oct. 1, 2020)

Abstract

This study mainly analyzed the theory of the attack-defense game. Firstly, it analyzed the game theory, then established the network attack-defense game model (NADGM), analyzed its replicator dynamics equation, expounded the network security game algorithm, and built the simulation experiment's network system. It was found from the results that the model and algorithm designed in this study could reflect the changes in network security well. It was also found from the analysis of nodes in the network that the number of infected nodes in the network first increased and then decreased. In contrast, the number of recovered nodes increased gradually, and the proportion of the final damaged nodes was about 10%. The experimental results verify that game theory is useful in network security and can be further promoted and applied.

Keywords: Attack-Defense Game; Defense Mechanism; Game Algorithm Network Security;; Replicator Dynamics

1 Introduction

With the rapid development of computer technology [11], the network has become an indispensable part of people's production and life [18] and played an irreplaceable role in various fields such as politics and the economy. Moreover, network security issues have become more prominent [13], and network attack means are becoming increasingly complex. Network security incidents may even affect the economic security of the world [9]. Therefore, network security has become an increasingly important issue. At present, most of the defense mechanisms used are passive and static [17], such as firewall [7], vulnerability scanning, etc., which can not comprehensively, accurately, and timely detect attacks and is not conducive to the establishment of dynamic defense mechanisms.

How to realize the dynamic and active defense of the network has become a research hotspot. Almohri *et al.* [2] designed a probabilistic graph model and algorithm to reduce the possibility of attacks on dynamic and complex networks, adopted the linear programming technology and verified the method's reliability by experiments on real large-scale networks. Mohamed *et al.* [1] studied the security of infinite sensor networks, designed an evolutionary game, a theory-based active defense model. They established a prevention mechanism to improve the stability and reliability of the network effectively.

Li *et al.* [12] designed a malicious code immune program based on an unbalanced support vector machine to achieve active defense against malware and maintain the system's stability in real-time. Guo *et al.* [10] studied Honeynet technology and provided the detailed implementation and deployment of solutions to realize a more effective security defense. In this study, the application of the game theory was studied. Based on the game theory, a model was established, and the algorithm was analyzed. The simulation experiment was carried out to verify the availability of the model in network security. This study makes some contributions to the realization of dynamic and reliable defense of the network.

2 The Establishment of Network Attack-Defense Game Model (NADGM)

2.1 Game Theory

Game theory was first applied in the field of the economy [15]. It mainly studies how the central bodies involved in decision-making make decisions. The interaction between the central bodies will change the decision making and equilibrium. The goal of all central bodies is to maximize interests [14]. There are three elements:

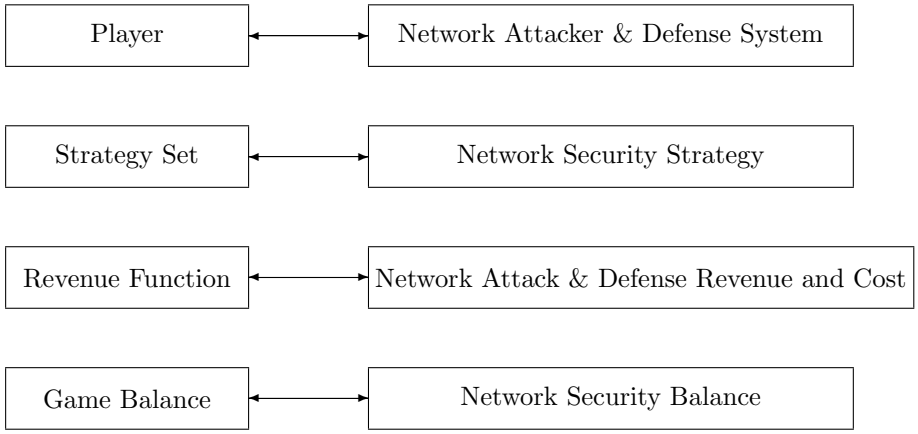


Figure 1: Network security and game theory

1) Player, i.e., decision-makers who will continuously seek the best strategy to maximize their interests;

2) The strategy set, i.e., the actions and rules of players; the more the strategies are, the more complex the game process is;

3) Revenue function, i.e., the profit obtained by players, which reflects the utility of different strategies.

Network security is also a process of the attack-defense game, as shown in Figure 1. Therefore, network security can be realized by establishing NADGM.

2.2 Modeling

NADGM can be described as a quadruple, $NADGM = (P, S, B, U)$. P is the set of players, $P = (P_A, P_D)$, where P_A is the attacker and P_D is the defender. S is the game action space, $S = (S_A, S_D)$, where S_A is the set of strategies of the attacker and S_D is the set of strategies of the defender. B is the set of game beliefs, $B = (B_A, B_D)$, where $B_A = \{B_{A_1}, B_{A_2}, \cdots, B_{A_i}\}$ and B_{A_i} stands for the probability of selecting attack strategy S_{A_i} $(i = 1, 2, \cdots, n)$. B_{D_j} represents the probability of selecting defending strategy S_{D_j}, $j = 1, 2, \cdots, m$. U is the set of revenue functions, $U = \{U_A, U_D\}$, where U_A is the attacker revenue function and U_D is the defender revenue function. The combination of those parameters can help calculate the revenue of attacker and defender, as shown in Table 1.

In Table 1, A_{ij} represents the revenue of attacker, $A_{ij} = DC(S_{A_i}) - DR(S_{D_j}) - AC(S_{A_i})$. DC represents the system loss cost; DR represents defense return; AC represents attack cost; D_{ij} represents the benefit of defender, $D_{ij} = DR(S_{D_j}) - DC(S_{A_i}) - DE(S_{D_j})$, and DE represents defending cost.

Table 1: Calculation of attack-defense revenue

Attacker	Expected return	$U_{A_i} = \sum_j^m B_{D_j} A_{ij}$
	Average revenue	$U_A = \sum_i^n B_{A_i} U_{A_i}$
Defender	Expected return	$U_{D_i} = \sum_j^n B_{D_j} D_{ji}$
	Average revenue	$U_D = \sum_i^m B_{D_i} U_{D_i}$

2.3 Replicator Dynamic Equation

It is assumed that there are $x_i(k)$ defenders selecting S_{D_i} at time k, which is $q_{D_i}(k)$ of the total, the expected return of S_{D_i} is $U_{D_i}(k)$, the average return is $\bar{U_{D_i}}(k)$, then the dynamic replicator equation can be written as follows:

$$q'_{D_i}(k) = q_{D_i}(k)(U_{D_i}(k) - \bar{U_{D_i}}(k)).$$

In the same way, the dynamic replicator equation of S_{A_i} can be written as:

$$q'_{A_i}(k) = q_{A_i}(k)(U_{A_i}(k) - \bar{U_{A_i}}(k)).$$

Combining the above two equations, there is:

$$Y = \begin{bmatrix} q'_{D_i}(k) \\ q'_{A_i}(k) \end{bmatrix} = 0.$$

The game equilibrium state point can be obtained by solving the above equation and the stable equilibrium solution.

2.4 Network Security Game Algorithm

The above model is applied to network security. It is assumed that the strategy set of the network defender is $S_D = \{S_{D_1}, S_{D_2}\}$, where S_{D_1} represents implementing network defense and S_{D_2} represents not implementing network defense). The strategy set of the attacker is $S_A = \{S_{A_1}, S_{A_2}\}$, where S_{A_1} represents implementing

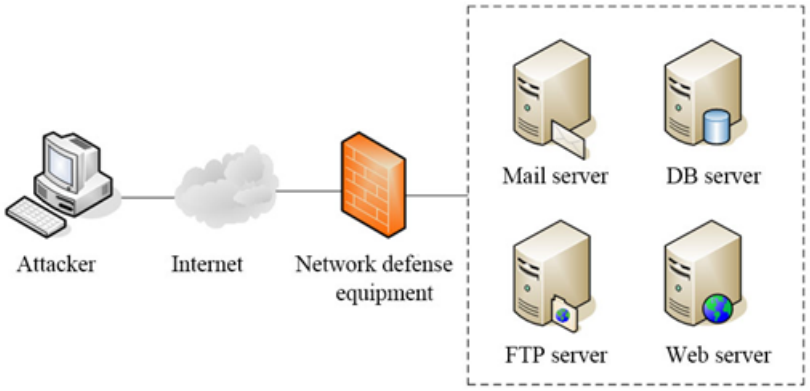


Figure 2: Experimental environment

network attacking and S_{A_2} represents not implementing network attacking. Then the attack-defense game tree can be obtained. In other words, when the attacker selected S_{A_1} with a probability of q_{A_1}, the defender may select S_{D_1} with a probability of q_{D_1} or select S_{D_2} with a probability of q_{D_2}; when the attacker selected S_{A_2} with a probability of q_{A_2}, it is similar.

The specific steps of the network security game algorithm are as follows.

1) The strategy set of the network attacker and defender is determined.

2) The revenue matrix is calculated, and the attack-defense game tree is established.

3) Probability inferences q_{A_i} and q_{D_i} of the attacker and defender are established.

4) The replicator dynamic equation of the attacker and defender is established;

5) The stable equilibrium solution is calculated by simultaneous equations;

6) The revenue of both sides in the stable state is calculated, and the attacking and defending strategies at that moment is output.

3 Experimental Analysis

A simple network information system was used for the experiment. The topological environment of the system is shown in Figure 2. There were1000 nodes in the system. The attacker was located in the external network. There were four servers in the internal network. The internal and external networks were separated by firewall equipment. The Web and Mail servers provided HTTP, SMTP, and IMAP services to the external network. DB and FTP servers could not be accessed from the external network. The attacker mainly regarded obtaining the root right of the FTP server to attack the internal network.

The network information system was scanned by a vulnerability scanning tool Nessus [6]. The vulnerability information is shown in Table 2.

Table 2: Vulnerability information

Hosts	Service Name	CVE number
Web server	Apache	CVE 2014-0098
DB server	Postgresql	CVE 2014-0063
FTP server	Linux	CVE 2013-1324
	MS-office	CVE 2014-0038

The possible network attacks were analyzed using MulVal open-source attack, and the network information system might have the following five states:

M1: Normal;

M2: Attack the Root right of web server;

M3: Aattack Root right of DB server;

M4: Attack User right of FTP server;

M5: Attack Root right of FTP server.

The NADGM model designed in this study was used for calculation. The attack path of the attacker was M1 → M2 → M3 → M4 → M5. Based on the MIT attack and defense behavior database [8], the optional attacks and defense strategies are shown in Tables 3 and 4.

According to the above attack and defense strategies, the stable equilibrium solutions under different states were analyzed, and then the revenues of both sides were calculated. The results are shown in Table 5.

The change curve of network security was analyzed, as shown in Figure 3.

It was seen from Figure 3 that the revenue of the attacker was more massive at the beginning. In this stage, the attacker gained the Web server's root permission through the vulnerability attack and obtained considerable revenue. At that moment, the defender has not

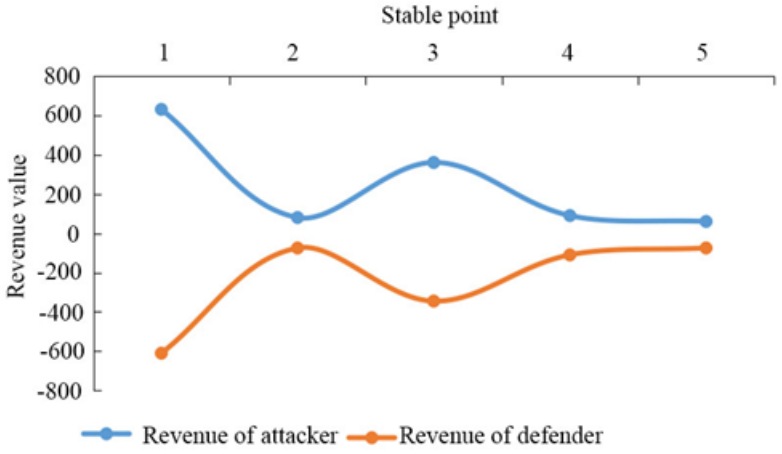


Figure 3: The change curve of network security

Table 3: Attack strategy

Number	Action name	Attack intensity
1	FTP host attack	0.3
2	LPC to LSASS	0.4
3	Send abnormal data to GIOP	0.5
4	Steal account and crack it	0.7
5	Install Trojan	0.8

Table 4: Defense strategy

Number	Action Name	Defensive Intensity
1	Patch SSH on FTP	0.2
2	Delete suspicious account	0.3
3	Repair database	0.4
4	Add physical resource	0.5
5	Restart database server	0.6
6	Renew root data	0.6
7	Uninstall delete Trojan	0.7
8	Reinstall listener program	0.8
9	Install oracle patch	0.8
10	Limit packets form ports	0.8

Table 5: Game results

State	Revenue of attacker	Revenue of defender
M1	630	-610
M2	80	-75
M3	360	-345
M4	90	-110
M5	60	-75

had time to make an adequate response. Then, a stable point 2, the defender began to detect and respond to the attack behavior; at that moment, the defender had increased revenue, while the attacker had decreased revenue.

A stable point 3, the attacker got the Root permission of the DB server, which caused a significant loss to the system; at that moment, the defender's revenue was less than that of the attacker. Then, at stable points 4 and 5, the revenue of both sides was stable. It was found that the NADGM model could reflect the changing trend of network security well.

If the Susceptible Infected Recovered (SIR) model was used for reference, the nodes in the system could be divided into four types:

N (normal node): The node in the normal working state;

I (infected node): The node that has been attacked but has no decline in the quality of service;

R (recovered node): The node which is under the protection of defense strategy and will not be damaged by the attack;

M (damaged node): The node has been attacked and has declined or completely lost service quality.

Table 6: Changes in the number of different nodes

Time point	Normal node	Infected node	Recovered node	Damaged node
0	1000	0	0	0
1	622	321	22	35
2	517	393	51	39
3	436	378	121	65
4	402	359	164	75
5	400	303	218	79
6	385	206	326	83
7	380	172	355	93
8	377	156	371	96
9	370	101	432	97
10	312	78	508	102

The change in the number of different nodes in the system was analyzed with the change of time. The results are shown in Table 6.

It was seen from Table 6 that the number of the normal nodes in the system gradually decreased; the number of the infected nodes increased first and then decreased. The number of recovered nodes and damaged nodes increased gradually with time. As the attacker implemented the attack strategy, many normal nodes in the system were attacked and became infected nodes in the initial stage. Then, as the defender's defense strategy, the number of the attacked nodes began to decrease, and the number of recovered nodes began to increase significantly. Finally, the number of nodes protected by the defense strategy was significantly larger than the number of the attacked nodes, indicating that the defense strategy had a healthy defense level. However, it can also be found from the changes of damaged nodes that the attacker's attack strategy has some influence on the system. About 10% of the nodes are damaged.

4 Discussion

Network security is a complex task [4]. The dynamic and active defense mechanism can realize the protection of the network better. The active defense can be carried out before the network crisis so that the attacker can not achieve its purpose. The active defense includes situation awareness [3], security detection [16], risk assessment [5], etc. This study mainly analyzed the attack-defense game model.

After analyzing game theory, this study designed the NADGM model, analyzed its dynamic replicator equation, described the network security game algorithm, and carried out simulation experiments to verify the method's performance. Through the construction of the simulation system and the simulation of attack and defense behaviors, it was found that network security could be accurately analyzed under the NADGM model and game algorithm designed in this study. When the attacker began to attack, the defender obtained a relatively large revenue as its response was not in time. After the defender began to detect and defend the attack, the revenue became more extensive than that of the attacker, indicating that the defender's strategy was effective.

As shown in Table 6, most of the system nodes were initially infected due to attacks. And their number is increasing. However, after the defense strategy began to take effect, the number of recovered nodes in the system increased rapidly, while the number of infected nodes decreased. It showed that the defense strategy played a useful role in protecting the system. Moreover, it was also found that about 10% of the nodes in the system were damaged because of the attack. Therefore, to realize the network security, the defender should:

1) Improve the speed of decision-making to make a timely and rapid response to attacks;

2) Increase defense investment and enhance defense intensity;

3) Scientifically arrange a defense strategy to enhance the defense effect.

5 Conclusions

In this study, the dynamic defense problem of network security was studied. Based on the attack-defense game theory, the NADGM model was designed, and the simulation system was built for test and analysis. It was found that:

1) The NADGM model can calculate the attack path of the attacker;

2) The NADGM model can understand the changing trend of network security;

3) In the network security game, the number of the attacked nodes increased and then decreased; with

the implementation of defense strategy, the recovered nodes increased, but about 10% damaged nodes.

The results showed that the NADGM model and game algorithm designed in this study were useful in network security. It could establish a dynamic defense mechanism and help the defender understand network situation changes and make an appropriate response.

References

[1] M. Al-Jaoufi, Y. Liu, Z. Zhang, "An active defense model with low power consumption and deviation for wireless sensor networks utilizing evolutionary game theory," *Energies*, vol. 11, no. 5, pp. 1281, 2018.

[2] H. M. J. Almohri, L. T. Watson, D. Yao, X. Ou, "Security optimization of dynamic networks with probabilistic graph modeling and linear programming," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 4, pp. 474-487, 2016.

[3] A. Alnusair, C. Zhong, M. Rawashdeh, M. S. Hossain, A. Alamri, "Context-aware multimodal recommendations of multimedia data in cyber situational awareness," *Multimedia Tools & Applications*, vol. 76, no. 21, pp. 1-21, 2017.

[4] N. Ben-Asher, C. Gonzalez, "Effects of cyber security knowledge on attack detection," *Computers in Human Behavior*, vol. 48, pp. 51-61, 2015.

[5] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, K. Stoddart, "A review of cyber security risk assessment methods for SCADA systems," *Computers & Security*, vol. 56, pp. 1-27, 2016.

[6] S. Chimmanee, T. Veeraprasit, C. Srisa-An, "A performance evaluation of vulnerability detection: NetClarity Audito, Nessus, and Retina," *International Journal of Computer Science & Network Security*, vol. 14, no. 3, pp. 34, 2014.

[7] N. H. Chowdhury, M. T. P. Adam, G. Skinner, "The impact of time pressure on cybersecurity behavior: A systematic literature review," *Behavior and Information Technology*, vol. 2019, pp. 1-19, 2019.

[8] L. A. Gordon, M. P. Loeb, W. Lucyshyn, R. Richardson, "CSI/FBI computer crime and security survey," in *Proceedings of the 2015 Computer Security Institute*, San Francisco, USA: IEEE Press, pp. 48-64, 2015.

[9] L. A. Gordon, M. P. Loeb, W. Lucyshyn, L. Zhou, "Externalities and the magnitude of cyber security underinvestment by private sector firms: A modification of the Gordon-Loeb model," *Journal of Information Security*, vol. 6, no. 1, pp. 24-30, 2015.

[10] H. Guo, J. Luo, Q. Geng, "A study on cyber defense honeynet technology and configuration examples," *International Journal of Simulation: Systems, Science & Technology*, vol. 17, no. 47, pp. 26.1-26.4, 2016.

[11] J. Li, J. Huang, L. Tian, J. Wang, "Application of new active defense technology in power information network security," *IOP Conference Series Materials Science and Engineering*, vol. 750:012156, 2020.

[12] P. Li, R. Wang, "Research on network malicious code immune based on imbalanced support vector machines," *Chinese Journal of Electronics*, vol. 24, no. 1, pp. 181-186, 2015.

[13] Y. Lv, Y. Guo, Q. Chen, G. Cheng, Y. Chen, "Active perceptive dynamic scheduling mechanism based on negative feedback," *procedia computer science*, vol. 131, pp. 520-524, 2018.

[14] K. Madani, M. Hooshyar, "A game theory–reinforcement learning (GT–RL) method to develop optimal operation policies for multi-operator reservoir systems," *Journal of Hydrology*, vol. 519, pp. 732-742, 2014.

[15] P. G. Palafox-Alcantar, D. V. L. Hunt, C. D. F. Rogers, "The complementary use of game theory for the circular economy: A review of waste management decision-making methods in civil engineering," *Waste Management*, vol. 102, pp. 598-612, 2019.

[16] F. Ullah, H. Naeem, S. Jabbar, S. Khalid, M. A. Latif, F. Al-Turjman, L. Mostarda, "Cyber security threats detection in internet of things using deep learning approach," *IEEE Access*, vol. 7, no. 99, pp. 124379-124389, 2019.

[17] J. Wei, R. Zhang, J. Liu, X. Niu, "Defense strategy of network security based on dynamic classification," *KSII Transactions on Internet & Information Systems*, vol. 9, no. 12, pp. 5116-5134, 2015.

[18] N. Zhu, "Research on network security model based on active and passive defense hybrid strategy," *Agro Food Industry Hi Tech*, vol. 28, no. 1, pp. 2686-2689, 2017.

Biography

Xuhua Zhao, born in 1983, graduated from Central South University in 2007. He has received a master's degree and is a senior engineer in Zhuhai City Polytechnic. He is interested in computer application, network security, and educational information.

Multi-format Speech Perception Hashing Algorithm Based on Short-Time Logarithmic Energy and Improved Mel Energy Parameter Fusions

Yi-Bo Huang[1], Yong Wang[1], Qiu-Yu Zhang[2], and He-Xiang Hou[1]
(Corresponding author: Yibo Huang)
College of Physics and Electronic Engineering, Northwest Normal University[1]
Anning District, Lan Zhou, China
School of Computer and Communication, Lanzhou University of Technology[2]
Qilihe District, Lan Zhou, China
(Email: huang_yibo@foxmail.com)
(Received June 19, 2019; Revised and Accepted Dec. 3, 2019; First Online Feb. 1, 2020)

Abstract

Aiming at the problems of single speech format, non-universal algorithm and low accuracy of tamper detection and location in existing speech content authentication algorithms, a multi-format speech perception hashing algorithm based on short-time logarithmic energy and improved energy parameter fusion is proposed. Firstly, the speech signal to be processed is preprocessed, and the short-time logarithmic energy and improved Mel energy of each frame are calculated. Then, perform time-frequency parameter fusion on time-frequency features by mean filtering, and the time-frequency parameters are constructed by difference hashing method. Finally, in order to improve the security of the algorithm, logical chaotic map is used to encrypt hash sequences with equal length scrambling. The experimental results show that the proposed algorithm not only has a good compromise between robustness and discrimination, but also has good robustness, discrimination and key dependence for multi-format speech signals, and can achieve small-scale tamper detection and location.

Keywords: Improved Mel Energy; Perceptual Hashing; Short-Time Logarithmic Energy; Speech Authentication; Tamper Detection and Localization

1 Introduction

Nowadays, with the popularization and diversification of computer network and information technology, the number of images, speech and video increase exponentially. As an important way of information transmission, the authenticity, integrity and security of speech signal become more and more important [4, 12, 21].

Traditional hashing technology is highly sensitive to multimedia content preservation operations, which makes traditional hashing technology no longer applicable to content authentication of speech. With the introduction of speech perception hashing technology [11], perceptual hashing technology with good robustness, discrimination and security has rapidly become one of the important methods of speech, video content authentication and security transmission. In recent years, the speech feature extraction methods based on perception hashing mainly include linear prediction-minimum mean square error (LP-MMSE) [13, 15], discrete cosine transform (DCT) [17], discrete wavelet transform (DWT) [16,20], Mel-frequency cepstral coefficients (MFCC) [1,6,7], and model [5,22], *etc.* Li *et al.* [8] and Chen et al. [2] proposed speech perception hashing algorithms with good robustness, but the discrimination is poor, the efficiency is not high, and the security problem is not considered. Zhang *et al.* [14] and Li *et al.* [9] proposed speech perception hashing algorithms with good robustness and efficiency, but poor discrimination. Chen *et al.* [3] proposed a perceptual hash algorithm with good anti-collision and robustness, but low efficiency. Li *et al.* [10] proposed a speech authentication algorithm with good robustness and security, which can achieve tamper detection and localization, but it has poor discrimination and low efficiency. The above algorithms are only for the study of single format speech signal, and they are not universal, and they do not have good anti-collision ability in content preservation operations. The multi-format speech perception hashing algorithm proposed by Zhang *et al.* [19], it achieves content authentication of five different speech formats including the original domain and the compressed domain. The

tamper detection and location accuracy is high, but the robustness and discrimination are poor, and the security issue is still not considered. The multi-format speech perception hashing algorithm proposed by Zhang *et al.* [18], it has high authentication efficiency. However, robustness and discrimination still need to be further improved.

In order to solve the above problems, a multi-format speech perception hashing algorithm based on time-frequency parameter fusion is proposed in this paper. The algorithm solves the problem that the algorithm is not universal, low accuracy of tamper detection and location in small-scale. It also has good robustness and discrimination to common speech formats WAV, MP3, FLAC, OGG and M4A.

2 Related Theory

2.1 Short-Time Logarithmic Energy

The energy of the speech signal changes with time, the energy difference between the unvoiced and voiced sounds is quite obvious. Therefore, the short-time energy is one of the most commonly used features in the time domain analysis of speech signals.

Let the speech signal be $x(n)$, and after pre-processing, the speech signal of the i-th frame is $y_i(n)$, $y_i(n)$ satisfies:

$$y_i(n) = w(n) * x((i-1) * inc + n) \quad 1 \le n \le L, 1 \le i \le N.$$

Where, $w(n)$ is a window function, generally a Rectangular window or a Hamming window. L is the frame length. inc is the frame shift length. N is the total number of frames after the frame.

The short-time energy ($E(i)$) formula of the i-th frame speech signal $y_i(n)$ is shown as Equation (1):

$$E(i) = \sum_{n=1}^{L} y_i^2(n) \quad 1 \le i \le N. \tag{1}$$

The short-time logarithmic energy (LE), which is an improved short-time energy after taking the logarithm, LE is shown as Equation (2):

$$LE(i) = \ln(E(i) + a) - \ln(a). \tag{2}$$

Where, a is a constant, $a = 1$.

2.2 Improved Mel Energy

The Mel frequency is to better simulate the auditory mechanism of the human ear, and convert the spectrum of the speech signal into the perceived frequency domain.

The energy of the sensing domain can be expressed by Mel energy (ME). The algorithm flow is shown in Figure 1. The ME is defined as:

$$ME(i) = \ln \sum_{m=1}^{M} s(i,m).$$

Where, $ME(i)$ is the i-th frame improved mel energy. $s(i,m)$ is the Mel subband energy of the m-th subband of the i-th frame, which is defined as:

$$s(i,m) = \sum_{k=1}^{N} E(i,k) * H_m(k) \qquad 1 \le m \le M \tag{3}$$

Where, $H_m(k)$ is the frequency response of the Mel filter. $E(i,k)$ is the energy spectrum of each frame (where i represents the i-th frame and k represents the k-th spectral line in the frequency domain), which is defined as:

$$E(i,k) = |x(i,k)|^2$$

Where, $x(i,k)$ is the fast Fourier transform of $y_i(n)$.

3 Construction of Multi-format Speech Perception Hash

The flow diagram of multi-format speech perception hashing algorithm based on short-time logarithmic energy and improved Mel energy parameter fusion is shown in Figure 2.

3.1 Perceptual Feature Extraction

Step 1: Pre-emphasis The input speech $x(n)$ is pre-emphasized to enhance the high frequency portion and obtain the pre-emphasis signal $x^{'}(n)$.

Step 2: Framing and windowing Firstly, divides $x^{'}(n)$ into a speech signal with a frame length of L, a frame shift of inc, and a total number of frames of N. Then the speech signal is smoothed by a Hamming window to obtain a speech signal $y_i(n)$, where $y_i(n)$ is the n-th sample value of the i-th frame.

Step 3: Short-term logarithmic energy Calculate the short-time energy$E(i)$ of the speech signal $y_i(n)$ of the i-th frame according to Equation (1). Then, calculate the short-time logarithmic energy of each frame according to Equation (2). The matrix of $LE(i)$ is shown as Equation (4):

$$LE = \begin{bmatrix} LE(1) & LE(2) & \cdots & LE(N) \end{bmatrix} \tag{4}$$

Step 4: Improved Mel Energy Calculate the Mel energy of the m-th subband of the i-th frame of the speech signal according to Equation (3), the matrix of $s(i,m)$ is shown as Equation (5). Then, calculate the ME of each frame of speech signal according to Equation (3), the matrix of ME is shown as Equation (6):

$$s(i,m) = \begin{bmatrix} s(1,1) & s(1,2) & \cdots & s(1,m) \\ s(2,1) & s(2,2) & \cdots & s(2,m) \\ \vdots & \vdots & \ddots & \vdots \\ s(i,1) & s(i,2) & \cdots & s(i,m) \end{bmatrix} \tag{5}$$

$$ME = \begin{bmatrix} ME(1) & ME(2) & \cdots & ME(N) \end{bmatrix} \tag{6}$$

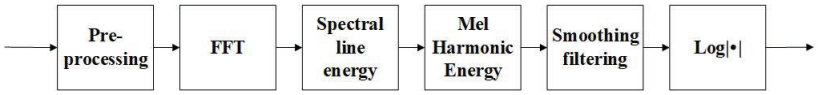


Figure 1: Improved Mel energy flow diagram

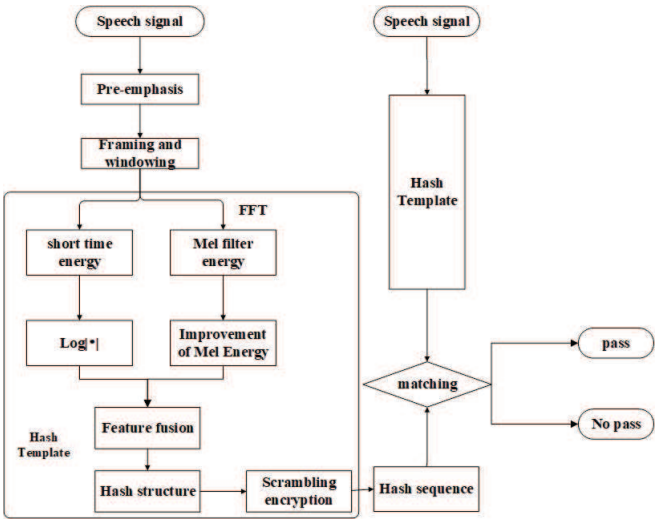


Figure 2: Speech perception hash authentication algorithm flow diagram

Step 5: Time-frequency parameter fusion Firstly, the time-domain parameter LE of each frame is multiplied by the frequency-domain parameter ME. Then, three-point mean filtering is performed to obtain the time-frequency fusion parameter $LME(LME((i)|i = 1, 2, \cdots, N))$, which is defined as:

$$LME(i) = smooth(c \cdot LE(i) \times ME(i)).$$

Where, *smooth* represents three-point mean filtering, c is a weighting factor, $c = 1$.

Step 6: Hash construction The LME of each frame is hash-structured by difference hash method to obtain hash sequence $h = h((i)|i = 1, 2, \cdots, N)$.

$$h = \left[\begin{array}{cccc} h(1) & h(2) & \cdots & h(N) \end{array} \right]$$

Constructing method: Set the hash sequence $h(1)$ to 0. If the i-th data of the vectorLME is greater than the $(i-1)$-th data, the i-th data of the hash sequence is 1, otherwise 0.

$$h(i) = \left\{ \begin{array}{ll} 1 \text{ if } h(i) > h(i-1) & \\ 0 \text{ otherwise} & (i = 2, 3, \cdots N) \end{array} \right\}$$

Step 7: Scramble the encryption Firstly, the logistic chaotic map is used to generate the pseudo-random sequence $s(i)(1 \le i \le N)$ with the same length as the hash sequence $h(i)$, the $s(i)$ satisfies Bernoulli independent distribution. Then, arrange $s(i)$ in descending order to obtain $s^{'}(i)$, there is a one-to-one mapping relationship between $s^{'}(i)$ and $h(i)$. Finally, by assigning $h(i)$ to $s^{'}(i)$ through the mapping relationship, and restoring $s^{'}(i)$ to the unsorted state, which forms a scrambling encryption for hash sequence $h^{'} = h^{'}((i)|i = 1, 2, \cdots, N)$.

3.2 Matching

During the authentication process, for the two speech segments x_1 and x_2, they will get the perceptual hash sequence h_1, h_2 after they pass the hash template. The normalized Hamming distance $D(:,:)$ of perceptual hash sequence can be regarded as bit error rate (BER), the calculation formula is as follows:

$$D(x_1, x_2) = d(h_1, h_2) = \frac{1}{N} \sum_{i=1}^{N} |h_1(i) - h_2(i)|$$

Where, D is the bit error rate.

In order to measure the overall performance of the algorithm, this paper uses the hypothesis test of BER to describe the hash match.

P_0: If the perceived content of the two speech segments x_1 and x_2 are the same, then: $D \le \tau$.

P_1: If the perceived content of the two speech segments x_1 and x_2 are different, then: $D > \tau$.

τ is the perceived authentication threshold. When the mathematical distance is less than or equal to the authentication threshold, it is passed, otherwise, it is not passed.

To further measure this algorithm, this paper defines the false accept rate (FAR) and false reject rate (FRR), which are:

$$R_{FAR} = \int_{-\infty}^{\tau} f(x|\mu,\sigma)d\alpha = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\tau} e^{\frac{-(x-\mu)^2}{2\sigma^2}} d\alpha.$$

$$R_{FRR} = \int_{-\infty}^{\tau} f(x|\mu,\sigma)d\alpha = 1 - \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\tau} e^{\frac{-(x-\mu)^2}{2\sigma^2}} d\alpha$$

Where, R_{FAR} represents the FAR, R_{FRR} represents the FRR, τ represents the perceived authentication threshold, μ represents mean of the BER, σ represents standard deviation of the BER.

4 Experimental Results and Analysis

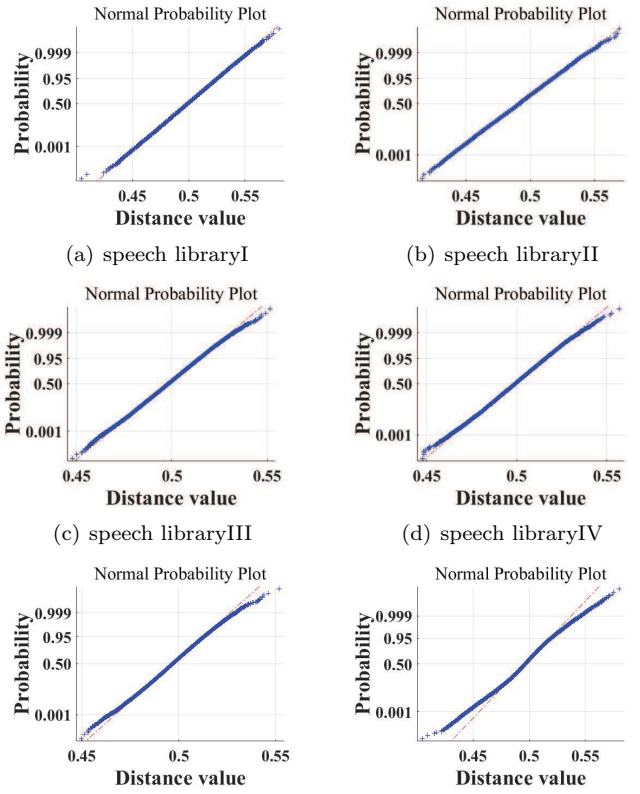
The speech data used in the experiment are speech signals in TIMIT(texas instruments and Massachusetts Institute of technology) and TTS(text to speech) speech libraries, in which the sampling frequency is 16kHz, the sampling precision is 16bit, the channel is mono-channel, the speech signal length is 4s. In order to verify the generality of the proposed algorithm for the authentication process of different speech formats, six speech libraries were established. Five of the speech libraries each contain a speech format (The formats of the speech libraries I, II, III, IV and V are WAV, MP3, FLAC, OGG and M4A respectively), each speech library is composed of 450 speech signals composed of Chinese men and women, English men and women, a total of 2250 speech signals. The other speech library TOATL is composed of the above five formats, a total of 2250 speech signals.

The experimental hardware platform is AMD A10-5750M CPU with Radeon(TM) HD Graphics, 4G, 2.5GHz, the software environment is MATLAB R2018b under Windows 10 operating system. The main parameters of the experiment are as follows:the frame length L = 256, the frame shift inc = 80, the number of frames N = 802, and the window function used is Hamming window.

4.1 Discrimination Analysis

Discrimination is used to evaluate the reliability of an algorithm in distinguishing between different or identical speech signal content. The BER of the perceptual hash value of different content speech signals basically obeys a normal distribution. From the BER data of each speech library obtained from the experiment, the normal distribution of the BER of each speech library can be obtained.

According to the De Moivre-Laplace central limit theorem, the Hamming distance approximation obeys a normal distribution($\mu = p, \sigma = \sqrt{p(1-p)/N}$), where p represents the probability of occurrence of 0 or 1 in the perceptual hash sequence, N represents the total number of



(a) speech libraryI (b) speech libraryII

(c) speech libraryIII (d) speech libraryIV

(e) speech libraryV (f) speech library TOTAL

Figure 3: BER distribution of the algorithm

frames.So the parameters of the Normal distribution in the ideal state are equal to $\mu = 0.5, \sigma = 0.0177$. The parameters of normal distribution obtained by each speech library in the experiment are as shown in Table 1.

Table 1: Normal distribution parameter values of each speech library

Speech library	Theoretical values		Experimental values	
	μ	σ	μ	σ
I	0.5	0.0177	0.4993	0.0178
II	0.5	0.0177	0.4911	0.0175
III	0.5	0.0177	0.4980	0.0110
IV	0.5	0.0177	0.4989	0.0114
V	0.5	0.0177	0.4968	0.0102

From Table 1, it can be seen that the normal distribution parameters of the proposed algorithm for eight different speech formats are very close to the theoretical values, so the algorithm has good randomness and anti-collision.

From Table 2 , when the frame shift of the algorithm is 128, 192 and 256 respectively, the FAR is about 5.9×10^5, 1.2×10^9 and 6.3×10^{10} times when the frame shift is 80. This is because when the speech segment is preprocessed, only the internal data of the window function is weighted, and the data outside the window is set

Table 2: Comparison of FAR of different frame shifting algorithm

τ	inc=80	inc=128	inc=192	inc=256
0.10	1.0232×10^{-112}	3.4775×10^{-71}	7.0784×10^{-48}	3.4588×10^{-36}
0.20	1.1250×10^{-64}	7.6298×10^{-41}	1.0299×10^{-27}	4.1983×10^{-21}
0.25	1.6400×10^{-45}	6.5425×10^{-38}	8.9089×10^{-20}	3.6614×10^{-15}
0.30	2.5396×10^{-29}	3.9529×10^{-19}	2.9067×10^{-13}	2.7695×10^{-10}
0.35	2.9210×10^{-17}	1.7193×10^{-11}	3.6493×10^{-8}	1.8527×10^{-6}

to 0. Therefore, as the frame shift is gradually increased, the overlapping frames between the speech segments are less, and the data are less, anti-collision ability also decreases.

As can be seen from the Table 2, FAR is also different when different frames, that is, in the same algorithm, FAR is affected by the length of the hash sequence, so in the experiment, it is not sufficient to measure the discrimination of the algorithm only by FAR. The entropy rate (ER) is a measure to measure the uncertainty of random events. It is also a good method to measure the discrimination of hash sequences. ER is defined as:

$$ER = -p\log_2 p - (1-p)\log_2(1-p),$$

where $p = \frac{1}{2}(\sqrt{(\frac{\sigma^2-\sigma_0^2}{\sigma^2+\sigma_0^2})}+1)$.

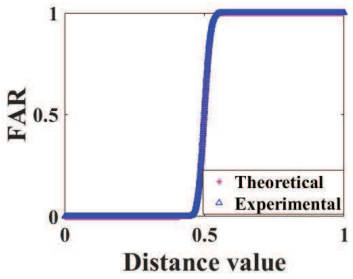


Figure 4: The FAR curve of the algorithm

Table 3: Entropy rate comparison of different algorithms

Algorithm	ER
Proposed algorithm	0.9957
Algorithm [10]	0.9864
Algorithm [3]	0.8633
Algorithm [16]	0.9187
Algorithm [15]	0.9745

From Figure 4 and Table 3, it can be seen that not only does the FAR experimental curve and the theoretical curve almost completely coincide, but the algorithm has a higher entropy rate, which further proves that the algorithm has good discrimination.

From Table 4, as the order increases, the FAR of the algorithm decreases, that is, the correct rate of the algorithm increases. This is due to the increase of the order, and the feature extraction points in the frequency domain also increase, which leads to the decrease of the algorithm's FAR .

It can be seen from Table 5 that compared with the other three algorithms, the proposed algorithm with overlapping frames has better anti-collision capability. Among them, when τ=0.35, the number of misjudged words per 1×10^{17} speech segments is 2.9210, and under the same conditions, it is 1.5×10^9 times smaller than [18], 2.9×10^{10} times smaller than [2], 3.4×10^{14} times smaller than [10].

4.2 Robustness Verification and Analysis

When performing the robustness test, eight kinds of content preservation operations shown in Table 6 are performed on five different speech libraries. Then, the average BER of the five different formats of the speech libraries is calculated after the content preservation operations is performed. The average BER of each speech library for content preservation operations is shown in Table 7.

According to the average value of each content preservation operations in Table 9, it can be seen that the average value of speech in different format speech librariess are mainly distributed in the region [0.0002,0.1750], which shows that the algorithm in this paper has good robustness to multi-format speech signals. Because this algorithm uses short-term energy and Mel energy as signal characteristics, combines with the various content preservation operations of five speech formats in Table 9, it can be concluded that for adding echo operation, because adding echo will superimpose the amplitude of speech signal and the amplitude of echo, so the robustness is poor. For narrowband noise operation, because the low signal-to-noise ratio has a greater influence on the probability density of the speech spectrum, so the robustness is poor. For the low-pass filtering operation, because of the filtering, the operation will filter out a part of the speech signal, so the robustness is poor. For resampling operations, the robustness is better because the signal amplitude and the speech spectrum are not changed. For the volume adjustment operation, since the WAV format and the MP3 format use lossy compression, which has a large influence on the speech spectrum, the WAV and the MP3 are less robust. The FLAC format and the M4A format have less influence on the speech spectrum, so FLAC and M4A have

Table 4: FAR of N-order algorithm

τ	8-order	16-order	24-order
0.10	1.2807×10^{-109}	1.2360×10^{-111}	1.0232×10^{-112}
0.20	1.7803×10^{-62}	1.2315×10^{-63}	1.1250×10^{-64}
0.25	6.0328×10^{-44}	6.1204×10^{-45}	1.6400×10^{-45}
0.30	9.0343×10^{-29}	2.6071×10^{-29}	2.5396×10^{-29}
0.35	6.1135×10^{-17}	2.9411×10^{-17}	2.9210×10^{-17}

Table 5: Comparison of FAR of different algorithms

τ	Proposed algorithm	Algorithm [18]	Algorithm [2]	Algorithm [10]
0.10	1.0232×10^{-112}	2.1420×10^{-47}	3.0310×10^{-38}	2.9390×10^{-12}
0.20	1.1250×10^{-64}	1.9220×10^{-27}	2.6890×10^{-22}	1.1440×10^{-5}
0.25	1.6400×10^{-45}	1.3754×10^{-18}	5.1740×10^{-16}	2.7150×10^{-4}
0.30	2.5396×10^{-29}	3.8423×10^{-13}	7.5420×10^{-11}	1.6820×10^{-3}
0.35	2.9210×10^{-17}	4.2761×10^{-8}	8.4900×10^{-7}	9.9900×10^{-3}

Table 6: Content preservation operations

Operation means	Operation method	Abbreviation
Volume adjustmean1	Volume up 50%	V.↑
Volume adjustmean2	Volume down 50%	V.↓
Resampling 1	Sampling frequency decrease to 8 kHz, and then increase to 16kHz	R.↑
Resampling 2	Sampling frequency increase to 32 kHz, and then droppsed to 16kHz	R.↓
Adding echo	Echo attenuation 25%, delay 300 ms	E.A
Narrowband noise	SNR=30 dB narrowband Gaussian noise, center frequency distribution in 0 4 kHz	G.N
Low-pass filtering1	12 order Butterworth low-pass filtering, Cutoff frequency of 3.4 kHz	B.W
Low-pass filtering2	12 order FIR low-pass filtering, Cutoff frequency of 3.4 kHz	F.I.R

Table 7: Average BER of each speech library after content preservation operations

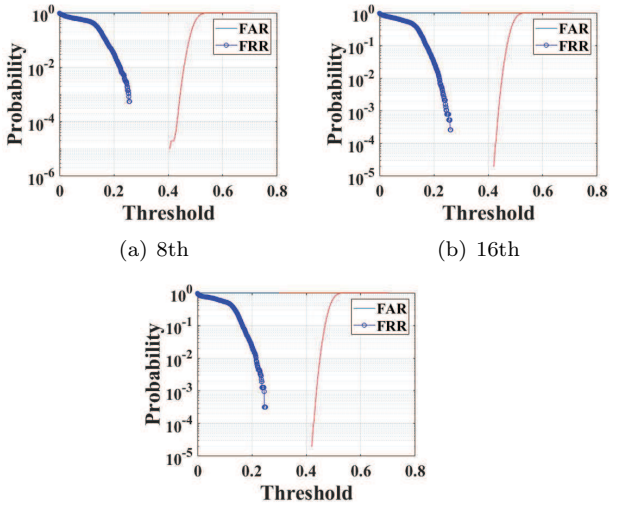
Operation means	I	II	III	IV	V
V.↑	0.1069	0.1551	0.0568	0.0643	0.0611
V.↓	0.1750	0.2021	0.0545	0.0633	0.0606
R.↑	0.0119	0.0811	0.0090	0.0113	0.0109
R.↓	0.0004	0.0351	0.0002	0.0017	0.0016
E.A	0.1669	0.2030	0.1506	0.1505	0.1490
G.N	0.1270	0.1666	0.1146	0.1146	0.1173
B.W	0.0711	0.1276	0.1393	0.1347	0.1347
F.I.R	0.0816	0.1331	0.1468	0.1408	0.1406

better robustness. Although OGG format is lossy compression, it is compressed by acoustic model, which has less influence on Mel energy feature extraction of analog acoustic model, so OGG format has better robustness.

It can be concluded from Figure 5 that as the order increases, the effect is getting better and better.

Figure 6 shows the FRR–FAR curve of [18], this paper uses five different formats of speech signals such as WAV, MP3, FLAC, OGG and M4A.

In the content preservation operations of the speech library, through the pairwise comparison of the perceived hash values, the speech libraries I – V each obtained 101025 data, the speech library TOTAL obtained 2530125 data, get the FAR and FRR of each speech library, and



(a) 8th (b) 16th

(c) 24th

Figure 5: N-order $FRR-FAR$ curves of the algorithm

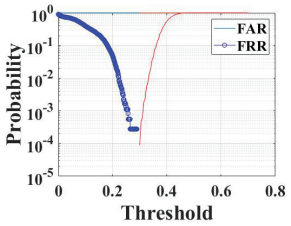
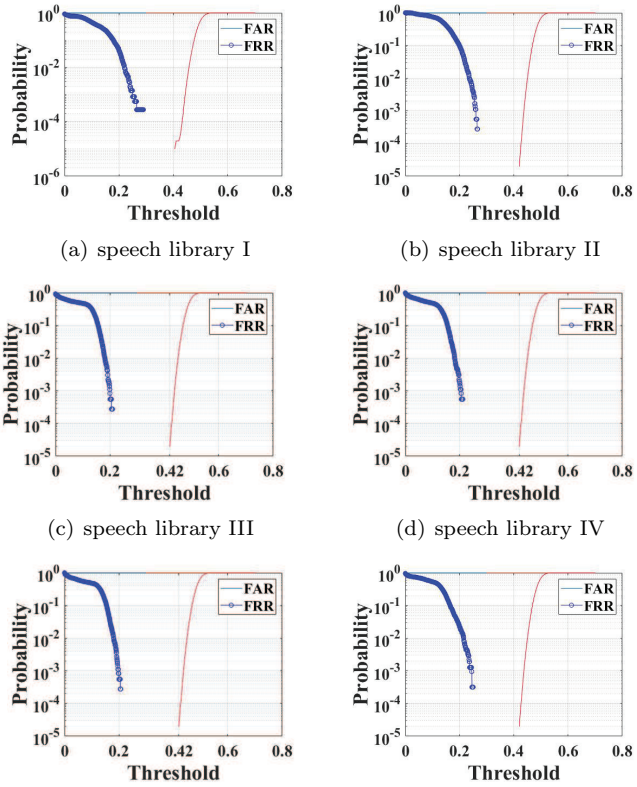


Figure 6: The $FRR-FAR$ curve of the algorithm [18]



(a) speech library I (b) speech library II

(c) speech library III (d) speech library IV

(e) speech library V (f) speech library TOTAL

Figure 7: $FRR-FAR$ curves for different speech libraries

the $FRR-FAR$ curves of Figure 7 is drawn according to the data.

By comparing Figure 6 with Figure 7, it can be concluded that although the proposed algorithm and [10] have good robustness, the $FRR-FAR$ curve decision interval of this algorithm is larger, that is, the compromise between robustness and discrimination is better.

In conclusion, the proposed algorithm has good robustness and discrimination.

4.3 Security Analysis

In order to improve the overall security performance of the proposed algorithm, a key-controlled logistic chaotic mapping algorithm is proposed.

In order to measure the disorder of the scrambling algorithm, the position number before scrambling and the change of the position number after scrambling are used to describe it in this paper.

T_0: If the position number before and after the scrambling of a speech segment has not changed, then:

$$\Delta_i = A(i) - A^{'}(i) = 0.$$

T_1 : If the position number before and after the scrambling of a speech segment changes, then:

$$\Delta_i = A(i) - A^{'}(i) \neq 0,$$

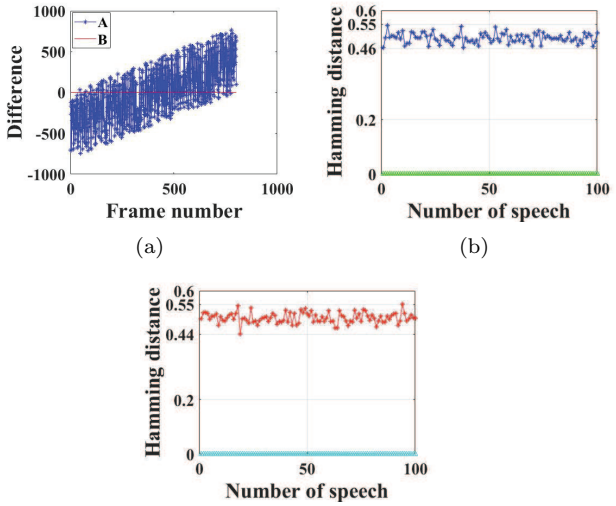
where Δ_i represents the position difference. $A(i)$ represents the i-th position number of the original hash sequence. $A^{'}(i)$ represents the i-th position number after scrambling.

It can be seen from Figure 8(a) that Δ_i and line $y = 0$ have very few intersections in the same sequence, this further proves the disorder of logistic chaotic maps.

In order to test the security before and after scrambling, this paper randomly extracts 100 speech segments from the speech library, calculates the Hamming distance of each frame before and after scrambling, and calculates the Hamming distance of the unencrypted hash sequence of the same speech after two feature extractions.

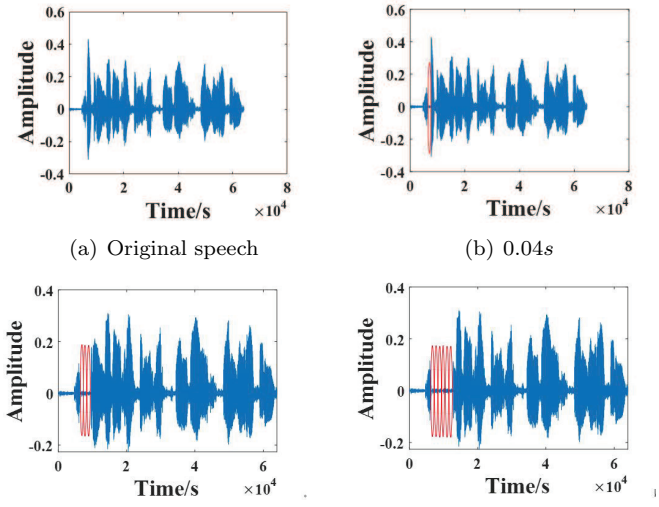
From Figure 7 and Figure 8(b), it can be seen that the decision interval of this algorithm is [0.2,0.42], the Hamming distance of each frame before and after scrambling is distributed in [0.46,0.55], the normalized Hamming codes of the unscrambled hash sequence are distributed on a straight line of $y = 0$. Therefore, the logistic chaotic map algorithm based on key control proposed in this paper has good security.

From Figure 8(c), it can see that the Hamming distance between the correct key hash sequence and the original hash sequence is distributed on the straight line $y = 0$, the hash sequence using the correct key and the Hamming distance using the wrong key are distributed in [0.44,0.55], which further proves that the proposed chaotic mapping algorithm has good security.



(a) (b) (c)

Figure 8: (a) The intersection of A:Difference of position number before and after scrambling and B:y=0. (b)Distribution of Hamming distance before and after speech scrambling in the same group. (c)Distribution of Hamming distance between correct key and error key.



(a) Original speech (b) $0.04s$ (c) $0.2s$ (d) $0.4s$

Figure 9: Tampering detection and location

To sum up, scrambling encryption does not change the size of Hamming distance of hash sequence, but only makes hash sequence disordered, thereby improving the security of the algorithm.

4.4 Tampering Detection and Location

For small-scale malicious attacks, it is generally tampering with the local part of the speech, the tampering range is small and the BER is low. This paper proposes a tamper detection and localization algorithm based on minimum code distance (MCD) of Hamming code.

In the process of tamper detection, for the original speech x and the tampered original speech $x^{'}$, the perceptual hash sequences $h(i)$ and $h^{'}(i)$ are obtained after passing through the hash sequence template. The MCD of the perceptual hash sequences $h(i)$ and $h^{'}(i)$ is defined as:

$$MCD(i) = \left\{ \begin{array}{l} 1 \text{ if } h(i) \neq h(i-1) \\ 0 \text{ otherwise} \end{array} \right\}$$

where, $MCD(i)$ is the MCD of the Hamming code of the i-th frame, and the matrix form of MCD is shown as Equation (7):

$$MCD(i) = \left[\begin{array}{cccc} MCD(1) & MCD(2) & \cdots & MCD(i) \end{array} \right] \quad (7)$$

In order to measure the small-scale tamper detection and localization ability of MCD algorithm, this paper randomly extracts a speech from the speech library. Since the speech duration is 4s, if it is 1%, 5%, 10% malicious attack, the duration of malicious attack should be separately $4s \times 1\% = 0.04s$, $4s \times 5\% = 0.2s$ and $4s \times 10\% = 0.4s$.

In Figure 9, blue represents speech and red regions represent tampering content regions. It can be concluded from Figure 9 that the algorithm in this paper can detect the tampering content area very well.

Table 8: Tampering detection and location in different ranges

Theoretical value	Experimental value
1%	1.30%
5%	5.20%
10%	10.20%

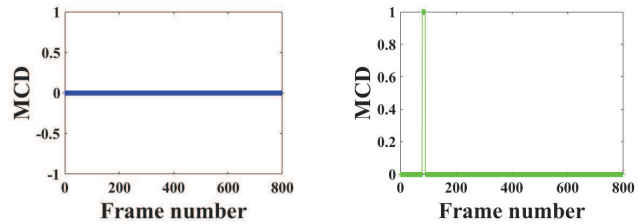
It can be seen from Figure 10 and Table 8 that the proposed algorithm not only can detect and locate the tampering area well, but also the actual attack range is very close to the theoretical attack range, both of which further prove that the algorithm has a good ability to tamper detection and localization.

To sum up, this algorithm can not only detect whether speech is maliciously attacked, but also locate small-scale tampering of speech.

4.5 Efficiency Analysis

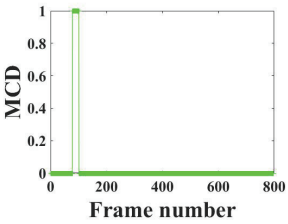
Efficiency analysis of the algorithm is an important factor to measure speech content authentication. Before calculating the computational efficiency of the proposed algorithm, 450 speech signal segments are randomly extracted from the speech library as the speech signal to be tested. Then the average running time of the algorithm is counted, it is shown as in Table 9. Finally, the algorithm is compared with single speech format the [2] and the [10] and multi-format speech format [18, 19], it is shown as Table 10.

From Table 9, it can be concluded that the efficiency of this algorithm is higher than that of the [2], but lower

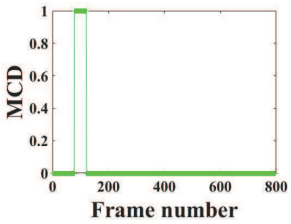


(a) Original speech

(b) 0.04s



(c) 0.2s



(d) 0.4s

Figure 10: The minimum code distance minimum code distance of Hamming code

Table 9: The average running time of the proposed algorithm in different frame shifts

Frame shifts	Overlap	N	Average running time/s
inc = 80	176	802	0.4333
inc = 128	128	501	0.3044
inc = 192	64	334	0.1844
inc = 256	0	251	0.1400

than that of [10, 18, 19]. Compared with the [3], this algorithm uses multi-feature extraction and has a large amount of computation. From Table 10, the average running time of the algorithm in different frame shifts can be concluded that the larger the frame shift, the smaller the overlapping frame, the faster the running efficiency. Compared with [18,19], this algorithm not only has overlapping frames, but also calculates Mel energy. Therefore, compared with [18, 19], which has simple structure and less feature extraction, the efficiency of this algorithm is lower. But this algorithm improves discrimination by adding overlapping frames. Moreover, this algorithm can satisfy the requirements of speech authentication for multi-format speech signals under real-time communication conditions.

5 Conclusions and Future Work

This paper presents a multi-format speech perception hashing algorithm based on short-time logarithmic energy and improved Mel energy parameter fusion. It not only solves the problems of the existing algorithms not universal, low anti-collision ability, low accuracy of tamper detection and location, but also has good robustness, discrimination and security. The experimental results show that the algorithm has good robustness to volume adjustment, resampling and low-pass filtering, has a good improvement in discrimination, and the algorithm has a good compromise between discrimination and robustness, has a great improvement in tamper detection and location, has a good enhanced in security, the Hamming distance before and after scrambling, the Hamming distance between the correct speech key and the wrong key are far from the decision domain. Because of the low frame shift in preprocessing, the efficiency of this algorithm is low. So we will improve it by studying frame shift in the next step.

Acknowledgment

This work is supported by the National Natural Science Foundation of China(No. 61862041), Youth Science and Technology Fund of Gansu Province of China(No. 1606RJYA274).

References

[1] A. Awais, S. Kun, Y. Yu, S. Hayat, A. Ahmed, and T. Tu, "Speaker recognition using mel frequency cepstral coefficient and locality sensitive hashing," in *International Conference on Artificial Intelligence and Big Data (ICAIBD'18)*, pp. 271–276, 2018.

[2] N. Chen, W. Wan, and H. D. Xiao, "Robust audio hashing based on discrete-wavelet-transform and non-negative matrix factorisation," *Iet Communications*, vol. 4, no. 14, pp. 1722–1731, 2010.

[3] N. Chen and H. D. Xiao, "Perceptual audio hashing algorithm based on zernike moment and maximum-likelihood watermark detection," *Digital Signal Processing*, vol. 23, no. 4, pp. 1216–1227, 2013.

[4] M. Dias, A. Abad, and I. Trancoso, "Exploring hashing and cryptonet based approaches for privacy-preserving speech emotion recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18)*, 2018. DOI: 10.1109/ICASSP.2018.8461451.

[5] Y. B. Huang and Q. Y. Zhang, "Strong robustness hash algorithm of speech perception based on tensor decomposition model," *Journal of Software Engineering*, vol. 11, pp. 22–31, 2017.

[6] Y. B. Huang, Q. Y. Zhang, and W. J. Hu, "Robust speech perception hashing authentication algorithm based on spectral subtraction and multi-feature tensor," *International Journal Network Security*, vol. 20, no. 2, pp. 206–216, 2018.

[7] E. Jokinen, R. Saeidi, T. Kinnunen, and P. Alku, "Vocal effort compensation for MFCC feature extraction in a shouted versus normal speaker recognition task," *Computer Speech & Language*, vol. 53, pp. 1–11, 2019.

Table 10: Comparing the operating efficiency of algorithms

Algorithm	Main frequency/GHz	Overlap	Average running time/s
Proposed Aalgorithm	2.5	176	0.4333
Aalgorithm [18]	3.2	0	0.0458
Aalgorithm [19]	2.5	0	0.0481
Aalgorithm [10]	2.5	0	0.4194
Aalgorithm [2]	3.2	0	0.5323

[8] L. I. Jinfeng, H. Wang, and Y. Jing, "Audio perceptual hashing based on NMF and MDCT coefficients," *Chinese Journal of Electronics*, vol. 24, no. 3, pp. 579–588, 2015.

[9] J. Li and T. Wu, "Perceptual audio hashing using RT and DCT in wavelet domain," in *The 11th International Conference on Computational Intelligence and Security (CIS'15)*, pp. 363–366, 2015.

[10] J. F. Li, T. Wu, and H. X. Wang, "Speech perception hash authentication algorithm based on MFCC correlation coefficient," *Journal of Beijing University of Posts and Telecommunications*, vol. 38, no. 2, pp. 89–93, 2015.

[11] X. M. Niu and Y. H. Jiao, "An overview of perceptual hashing," *Acta Electronica Sinica*, vol. 36, no. 7, pp. 1405–1411, 2008.

[12] I. Siddavatam, D. Khatri, P. Ashar, V. Parekh, and T. Sharma, "Authentication using dynamic question generation," in *Integrated Intelligent Computing, Communication and Security*, pp. 293–300, 2019.

[13] Q. Y. Zhang, W. J. Hu, Y. B. Huang, and S. B. Qiao, "An efficient perceptual hashing based on improved spectral entropy for speech authentication," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 1555–1581, 2018.

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

[15] Q. Y. Zhang, W. J. Hu, S. B. Qiao, and T. Zhang, "An efficient speech perception hash authentication algorithm based on the linear prediction minimum mean squared error," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 44, no. 12, pp. 127–132, 2016.

[16] Q. Y. Zhang, S. B. Qiao, Y. B. Huang, and T. Zhang, "A high-performance speech perceptual hashing authentication algorithm based on discrete wavelet transform and measurement matrix," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 1–17, 2018.

[17] Q. Y. Zhang, S. B. Qiao, T. Zhang, and Y. B. Huang, "A fast speech feature extraction method based on perceptual hashing," *International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD'17)*, pp. 1295–1300, 2017.

[18] Q. Y. Zhang, S. B. Qiao, T. Zhang, and Y. B. Huang, "Perception hashing authentication algorithm for multi-format audio based on energy to zero ratio," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 45, no. 9, pp. 33–38, 2017.

[19] Q. Y. Zhang, P. F. Xing, Y. B. Huang, R. H. Dong, and R. H. Yang, "Perception hashing algorithm for multi-format audio," *Journal of Beijing University of Posts and Telecommunications*, vol. 39, no. 4, pp. 77–82, 2016.

[20] Q. Y. Zhang, P. F. Xing, Y. B. Huang, R. Dong, and Z. Yang, "An efficient speech perceptual hashing authentication algorithm based on DWT and symmetric ternary string," *International Journal of Information and Communication Technology*, vol. 12, no. 1/2, pp. 31, 2018.

[21] X. Zhang, J. Zhang, T. He, Y. Chen, Y. Shen, and X. Xu, "A speech and lip authentication system based on android smart phone," in *Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*, pp. 110–114, 2018.

[22] Q. Y. Zhang, T. Zhang, D. F. Wu, and Z. X. Ge, "Strong robust speech authentication algorithm based on quasi-harmonic model," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 46, no. 3, pp. 58–64, 2018.

Biography

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

Wang Yong received the BS degrees in Henan Institute of Science and Technology, Henan, China, in 2017. His research interests include audio signal processing and application, multimedia authentication techniques.

Zhang Qiu-yu researcher/Ph.D. supervisor, graduated from Gansu University of Technology in 1986, and then worked at school of computer and communication in Lanzhou University of Technology. He is vice dean of Gansu manufacturing information engineering research

center, a CCF senior member, a member of IEEE and ACM. His research interests include network and information security, information hiding and steganalysis, multimedia communication technology.

Hou He-Xiang received the BS degrees in communication engineering from Dezhou University, Shandong, China, in 2018. His research interests include audio signal processing and application, multimedia authentication techniques.

Reviewers (Volume 22, 2020)

Dariush Abbasinezhad
Slim Abdelhedi
Mohd Faizal Abdollah
Ahmed Mohammed Abdullah
Subrata Acharya
Sodeif Ahadpou
Tohari Ahmad
Muhammad Najmi
Ahmad-Zabidi
Mohammad Reza Ahmadi
Asimi Ahmed
Mehrnaz Akbari Roumani
Abdul-Gabbar Tarish
Al-Tamimi
Aws N. Al-Zarqawee
Monjur M Alam
Shahid Alam
Tanweer Alam
Dilip S Aldar
Sara Ali
Ali Mohamed Allam
Khalid Abdulrazzaq
Alminshid
Ali Mohammed Alsahlany
Ruhul Amin
Rengarajan Amirtharajan
R. Anand
Karl Andersson
Benjamin Arazi
K. S. Arvind
Travis Atkison
Hany Fathy Atlam
Cossi Blaise Avoussoukpo
Anant M. Bagade
Amandeep Bagga
Nischay Bahl
Anuj Kumar Baitha
Saad Haj Bakry
R. R. Balakrishnan
Kavitha Balu
Maram Y Bani Younes
Tamer Mohamed Barakat
Utpal Barman
Pijush Barthakur
Eihab Bashier Mohammed
Bashier
Adil Bashir
Sunny Behal
Rydhm Beri
Taran Singh Bharati
Akashdeep Bhardwaj
Lathies T. Bhasker
Sugandh Bhatia
Sajal Bhatia
Krishna Bhowal
Sumitra Binu
Zhengjun Cao
Liling Cao
Chi-Shiang Chan
Eric Chan-Tin
Mohan Kumar Chandol
Yogesh Chandra
Arup Kumar Chattopadhyay
Nirbhay K. Chaubey
Ali M Chehab
Chi-Hua Chen
Tzung-Her Chen
Zhixiong Chen
Yi-Hui Chen
Chin-Ling Chen
Jan Min Chen
Qingfeng Cheng
Kaouthar Chetioui
Mao-Lun Chiang
Shu-Fen Chiou
Tae-Young Choe
Kim-Kwang Raymond Choo
Christopher P. Collins
Joshua C. Dagadu
Ashok Kumar Das
Prodipto Das
Sanjoy Das
Debasis Das
Ranjan Kumar Dash
Subhrajyoti Deb
Abdelrahman Desoky Desoky
Sankhanil Dey
Subhasish Dhal
Jintai Ding
Nishant Doshi
Ahmed Drissi
Qi Duan
Abd Allah Adel Elhabshy

Ahmed A. Elngar
Arizona Firdonsyah
Xingbing Fu
Vladimir Sergeevich Galyaev
Rakesh C Gangwar
Juntao Gao
Tiegang Gao
Xinwei Gao
G. Geetha
Mohammad GhasemiGol
Madhumala Ghosh
Ramesh Gopalan
Poornima Ediga Goud
Krishan Kumar Goyal
Ke Gu
Sumalatha Gunnala
Jatin Gupta
Charifa Hanin
Ali Hassan
Wien Hong
Tsung-Chih Hsiao
Defa Hu
Yen-Hung Hu
Xiong Hu
Chengyu Hu
Huajun Huang
Chin-Tser Huang
Jianmeng Huang
Munawar Hussain
Bala Venkateswarlu Isunuri
Grasha Jacob
Amit Jain
Yogendra Kumar Jain
Swati Jaiswal
Teena Jaiswal
V. S. Janani
N Jeyanthi
lin zhi jiang
Shaoquan Jiang
Rong Jiang
Rui Jiang
Zhengping Jin
Ashish Joshi
Omprakash Kaiwartya
Yoshito Kanamori
Nirmalya Kar
Gagandeep Kaur
Omar Khadir
Vaishali D. Khairnar
Asif Uddin Khan
Md. Al-Amin Khandaker
Malik Sikander Hayat Khiyal
Dong Seong Kim
P. Dhandapani Raman D. Kothandaraman
Anjan Krishnamurthy
Sajja Ratan Kumar
Manish Kumar
Naresh N Kumar
Saru Kumari
Yesem Kurt Peker
Then Lee
Yanping Li
Chun-Ta Li
Cheng Li
Zhaozheng Li
Chia-Chen Lin
Chih-Yang Lin
Iuon-Chang Lin
Yang-Bin Lin
Yining Liu
Shuang Gen Liu
Ximeng Liu
K. Shantha Kumari Luke Jayakumar
Zhiyong Luo
Ming Luo
Zahid Mahmood
Tanmoy Maitra
Arun Malik
T. Manesh
Ali Mansouri
Kamran Ali Memon
Weizhi Meng
Bo Meng
Yang Ming
Suhail Qadir Mir
Amit Mishra
anuranjan Misra
Madihah Mohd Saudi
Guillermo Morales-Luna
Belmekki Mostafa
Alaa Moualla
Hamdy M. Mousa
Kuntal Mukherjee
C. H. Mukundha
Bhagavathi Priya M Muthumanikam
Ambika Nagaraj
K. Nandhini
Syed Naqvi
Lakshmi Kannan Narnayanan

Sarmistha Neogy
Chokri Nouar
Abdul Abiodun Orunsolu
Nasrollah Pakniat
Dhiraj Pandey
B. D. Parameshachari
Subhash S. Parimalla
Chintan J. Patel
Kailas Ravsaheb Patil
Suresh Kumar Peddoju
Kanthakumar Pongaliur
A. Prakash
Munivara Prasad
Yudha Purwanto
Septafiansyah Dwi Putra
Murad Abdo Rassam Qasm
Qais Saif Qassim
Chuan Qin
Jiaohua Qin
Narasimhan Renga Raajan
Hashum Mohamed Rafiq
Abdul Hamid M. Ragab
V. Sampangi Raghav
Uma R. Rani
Golagani A.V.R.C Rao
Dhivya Ravi
Ramesh S Rawat
Siva Ranjani Reddi
Ou Ruan
Sanjay Kumar Sahay
Debabrata Samanta
Sabyasachi Samanta
Manju Sanghi
Arif Sari
Balamurugan K. S. Sathiah
Rajat Saxena
Michael Scott
Chandra Vorugunti Sekhar
Irwan Sembiring
Elena Sendroiu
Divyashikha Sethia
Vrutik M. Shah
Vrushank Shah
Kareemulla Shaik
Tarun Narayan Shankar
Udhayakumar Shanmugam
Rohith Shivashankar
Varun Shukla
Jitendra Singh
Debabrata Singh
Anuj Kumar Singh
Mahendra Pratap Singh
Bala Srinivasan
Siva Shankar Subramanian
Karthikeyan Subramanian
T. SudalaiMuthu
K. S. Suganya
Haiyan Sun
Maryam Tanha
Ariel Soares Teles
Pratik Teli
Xiuxia Tian
Geetam Singh Tomar
Yuan-Yu Tsai
Vandani Verma
Phu Vo Ngoc
Tao Wan
Fangwei Wang
Li Wang
Feng Wang
Libin Wang
Ying Wang
Ding Wang
Qingping Wang
Zhe Wei
C. H. Wei
Jianghong Wei
Na-I Wu
Degang Xu
Lei Xu
Chengbo Xu
Yashveer Yadav
Wei Yajuan
Li Yang
Wenjie Yang
Changsong Yang
Yifei Yao
Jun Ye
Pinghao Ye
Huang Yiwang
Lin You
Huifang Yu
Hang Yue
Dr Noor Zaman Zaman
Sherali Zeadally
Jianping Zeng
Jie Xiu Zhang
Qiu-Yu Zhang
Yanshuo Zhang
Fangguo Zhang
Zonghua Zhang
Futai Zhang

Yinghui Zhang
Jianhong Zhang
Hongzhuan Zhao
Zhiping Zhou
Ye Zhu
Yingwu Zhu
Frank Zhu
Aaron Zimba

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.