Security and Communication Networks

Privacy-Enhancing Authentication and Computation for Human CPS and IoT

Lead Guest Editor: Yang Zheng
Guest Editors: Rongxing Lu, Jianting Ning, Zengpeng Li, and Sridhar Adepu

Privacy-Enhancing Authentication and Computation for Human CPS and IoT

Security and Communication Networks

Privacy-Enhancing Authentication and Computation for Human CPS and IoT

Lead Guest Editor: Yang Zheng
Guest Editors: Rongxing Lu, Jianting Ning, Zengpeng Li, and Sridhar Adepu

Contents

Hindawi
Security and Communication Networks
Volume 2023, Article ID 9621839, 11 pages
https://doi.org/10.1155/2023/9621839

Research Article

Blockchain-Based Privacy-Preserving Sensor Data Sharing with Fine-Grained Authorization in Microgrid

Jinhu Yu, Yue Han, Kai Zhang, Siyuan Chen, and Jinguo Li

College of Computer Science and Technology, Shanghai University of Electric Power China, Shanghai, China

Correspondence should be addressed to Jinguo Li; lijg@shiep.edu.cn

Received 28 July 2022; Revised 7 September 2022; Accepted 3 October 2022; Published 11 May 2023

Academic Editor: Zheng Yang

Microgrid is a power system that includes various energy sources (e.g., solar panels and wind turbines), where a number of device status and sensing data are collected and transmitted by smart sensors. Based on sensing-as-a-service in microgrid, sensor owners and sensor data consumers can effectively perform data sharing operations. However, the state-of-the-art sensor data sharing works in microgrid have the following two limitations: (i) cannot support fine-grained authorization for sensor owners and sensor data consumers and (ii) fail to simultaneously consider confidentiality and authenticity for sensor data sharing. To address the problems, in this article, we propose a lightweight privacy-preserving sensing data sharing system with fine-grained authorization in microgrid. Technically, we employed attribute-based signature methodology to design a fined-grained authorization mechanism for sensor data users. Moreover, a lightweight hyper elliptic curve-based signcryption scheme is employed to provide confidentiality and authenticity for sensor data sharing. To clarify the feasibility of our proposed system, we implement the system and evaluate the performance. The experimental results show that the system achieves small communication and time overhead, as well as highly acceptable gas consumption of smart contract.

1. Introduction

With the access of multiple energy sources and numerous power loads, the traditional power system is rapidly evolving into a microgrid [1]. Specifically, a microgrid is a self-sufficient power system that includes distributed power sources, energy storage devices, transmission grids, and user loads. Due to the rapid development and wide employment of 5G and Internet of Things (IoT) [2], the microgrid system can perform measurement operations (e.g., data collection, data transmission, and data analysis) based on smart IoT devices. Hence, the information of microgrid operation status, equipment status, and energy data are effectively sensed and monitored by these smart sensors, which provides a guarantee of safe operation for microgrid. According to an IHS analysis, the smart grid or microgrid-related sensor market has grown nearly tenfold between 2014 and 2021, reaching 350 million dollars, which is expected that there will be 41.6 billion IoT sensing devices by 2025.

There are amounts of sensors and connected devices that is deployed in microgrid, which is followed by the large-scale data perception and processing tasks. This motivates the employment of Sensor-as-a-Service (SaaS) [3] into the microgrid (as illustrated in Figure 1), which is driven and influenced by cloud computing service. In SaaS, sensor owner can collect, packet, and process sensing data, thus data consumers can reuse and acquire sensing data with relatively low cost. As a result, sensor owners and data consumers can securely and effectively perform data sharing and trading [4]. Since the fairness of data sharing and transactions in traditional SaaS model only relies on the service provider. Once the trust of service providers is lost, the security and fairness of data sharing may become a serious challenge.

In data sharing service, the primary security goal and fairness is to ensure that the real identities of users and transmitted data are not leaked, and third-party involvement should be avoided as far as possible in the transaction process to maximize the interests of both parties. To achieve data privacy and fairness, blockchain [5] was introduced into

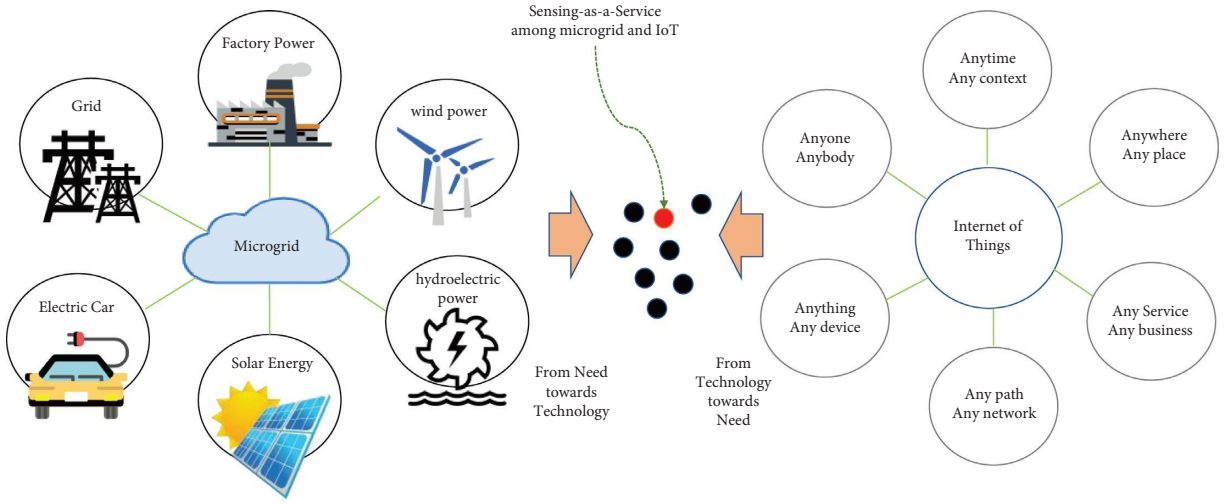


Figure 1: Relationship among sensing as a service model, microgrid, and internet of things.

data sharing services. Blockchain is a decentralized platform that combines peer-to-peer networks, cryptographic protocols, distributed data storage, and consensus mechanisms, where a number of transactions that used for recording data interaction is created and packaged into a block and added to the blockchain by miners. After that, every peer node can verify the validity of data transactions through cryptographic algorithms and consensus protocols. Due to these positive characteristics, blockchain technology has been extensively researched and deployed in practical data sharing services [4, 6–9].

The recent proposed blockchain-based data sharing systems [6, 7, 9] considered different properties of privacy-preserving in data sharing services, such as fairness, anonymity, and traceability. To enable users more willing to participate in data sharing, Samuel et al. [8] combined access control module with differential privacy and thus gave a blockchain-based fair data sharing for deregulated smart grids. By combining the advantages of IoT and SaaS in smart city, Lin et al. [4] presented an effective blockchain-based data sharing system based on symmetrical encryption and signature, Paillier encryption, and Σ-protocol. Nevertheless, these blockchain-based data sharing systems cannot be directly used in microgrid driven by sensing-as-a-service. This is because of the following reasons:

(i) The sensor devices in microgrid are usually equipped with more constrained computation and storage resources

(ii) The number of sensor owners and data consumers are large that require fine-grained access control strategies

(iii) The shared sensor data is usually provided with either data confidentiality guarantee or data authenticity guarantee

1.1. Our Results

1.1.1. Motivation. To address the problems, we proposed an effective blockchain-based sensor data sharing system in microgrid, which considers practical efficiency and security requirements. Generally, the service provider can perform fine-grained authorization over data user registration and achieve lightweight enhanced privacy-preserving of data confidentiality and data authenticity for the shared sensor data.

In particular, the contributions of this work can be summarized as follows:

(1) Fine-grained authorization. To enable the system to perform fine-grained authorization to sensor owner and data consumers, we design a fine-grained authorization mechanism based on attribute-based signature for user registration. In particular, a registered user is granted with a corresponding number of pseudonyms. As a result, not only the real identity of the user is preserved, but also the fine-grained access control of user registration is realized.

(2) Enhanced privacy-preserving data sharing. To provide enhanced privacy guarantee for the blockchain-based data sharing platform, we use a lightweight hyper elliptic curve-based signcryption scheme to achieve both confidentiality and authenticity for the shared sensor data. In particular, we employed a key encapsulation mechanism into our sensor data sharing system, where the sensor data is encrypted by AES and the key of AES is signcrypted by the lightweight hyper elliptic curve-based signcryption scheme.

(3) Reputation-based sensor owner selection. To prevent much manual intervention for a sensor owner selection, we employed a blockchain platform with constructing an effective and efficient sensor owner selection model based on reputation calculation. In particular, we designed smart contracts and formulate a reputation calculation function for each sensor owner, where the function considers the following factors, such as transaction frequency,

positive and negative reviews, and the real-time nature of reviews.

In addition, we write the codes and deploy the corresponding smart contracts in Remix Rem. Particularly, we designed 8 functions in smart contracts and evaluate their gas cost, where the cost is all below 2.0×10^6Gwei. Moreover, we evaluate the communication cost and computational overhead of AES, attribute-based signature, and signcryption that are used in our proposed system, where the cost is highly acceptable due to simple AES and lightweight hyper elliptic curve-based signcryption scheme.

1.1.2. Organization. Section 2 reviews some background knowledge and Section 3 formalizes the system model and security requirements. In Sections 4 and 5, we presented the construction and security analysis of the proposed sensor data sharing system in microgrid. Section 7 surveys recent related works and Section 8 finally concludes this work.

2. Preliminaries

2.1. Blockchain and Smart Contract. Blockchain is a decentralized distributed ledger that can record, store, and update data in a distributed manner. Transactions, in a blockchain, are the most basic activities that miners create, record, and approve in a block. Miners who with accounting rights send the blocks they create to each peer node in the system via a consensus algorithm. When received by other nodes, blocks are verified for hash, signature, and transaction validity, and after the consensus is formed, they are added locally. Furthermore, when the preparatory conditions are met, smart contracts execute, which are stored on the blockchain. They typically act as protocols enforced by specific rules that are predefined by computer code and replicated and enforced by all network nodes.

2.2. Attribute-Based Signature. Li et al. [10] initiates the notion of attribute-based signature (ABS), in which a sensor owner can sign messages with any policy that composed up of a number of attributes. Correspondingly, only the specified policy is revealed to the public while the user's identity is kept in privacy.

(1) ABS.Setup: This algorithm takes a security parameter λ as input and generates a public parameter PP and a master key MK.

(2) ABS.KeyGen: This algorithm takes PP and MK and a data user's attributes Γ as inputs and generates a private key SK_Γ for the user.

(3) ABS.Sign: This algorithm takes PP, a message M, a data user's SK_Γ, a policy Λ that accepts Γ, and finally signs the message M to output a signature δ.

(4) ABS.Verify: This algorithm takes PP and δ and attributes Γ as inputs, and outputs 1 if δ is a valid signature.

2.3. A Signcryption Scheme Based on Hyper Elliptic Curve. The work in [11] gave a highly efficient signcryption scheme based on hyper elliptic curve, where the signcryption algorithm and unsigncryption algorithm are described as follows:

(i) Signcryption (k, d_a, m, P_b, P_a)

(a) Randomly selects an integer $k \in [1, n-1]$
(b) $(K_1) = h(\phi(kD))$
(c) $(K_2) = h(\phi(kD))$
(d) $C = E_{K_2}(m)$
(e) Compute $r = h_{K_1}(\mathrm{m} \| \text{bind info})$
(f) Compute $s = (K/(r + d_a)) \mod n$
(g) Compute $R = rD$
(1) Thus, the signcrypted transmitted text is (c, R, s).

(ii) Unsigncryption $(P_b, P_a, d_b, h, c, R, s)$

(a) Compute (K_1, K_2)
(b) $(K_1) = H(\phi(s(P_a + R)))$
(c) $(K_2) = H(\phi(s(d_b(P_a + R))))$
(d) Compute $m = D_{K_2}(c)$
(e) Compute $r \stackrel{?}{=} h_{K_1}(\mathrm{m} \| \text{bind info})$
(f) Check $rD \stackrel{?}{=} R$, if true accept the message, else reject.

3. Problem Formulation

In this section, we formalized the system model and security requirements for our proposed sensor data sharing system in microgrid.

3.1. System Model. There are three main entities that is considered in our proposed data sharing system: sensor owners, data consumers, and service providers, as shown in Figure 2. In particular, the formal descriptions of these entities are as follows:

(1) Sensor owners: Sensors usually refer to devices that are connected to various energy equipments for measuring, sensing, and presenting data information (e.g., temperature, humidity, and electricity). In particular, the sensors can satisfy the requirements of information transmission, processing, storage, display, recording, and its characteristics, such as miniaturization, intelligence, networking, and other characteristics. Generally, the sensor owners are independent parties who have these sensors in possession and a sensor owner may own one or more sensors. If the sensor owner is willing to share the data in the sensor, paid, or free, then they can publish the sales information in the system.

(2) Data consumers: Data consumers (e.g., energy companies, scientific research teams, and schools) may purchase the sensing data by Saas model. In the system, data consumers can send requests for

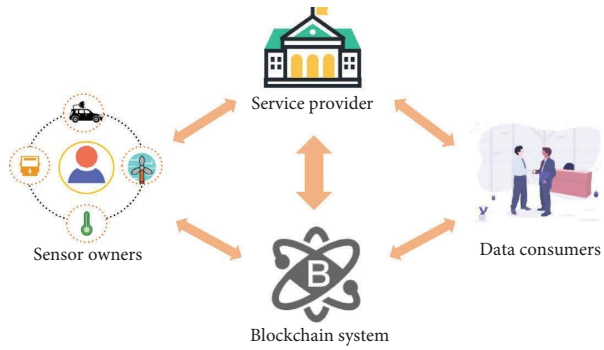


Figure 2: System model of data sharing.

matching candidate sensor data that is contributed by different sensor owners. If data consumers intend to purchase shared sensor data, they may pay a deposit to the sensor owner in advance, and then pay the corresponding balance after obtaining all sensor data. Moreover, all transaction processes are automatically completed via the smart contracts in the system.

(3) Service provider: The honest and curious service provider runs registration service for sensor owners and data consumers, where only the registered parties can conduct data transactions in the system. Note that the registration service is completed based on the deployed smart contracts in the system. In addition, the service provider stores the data to be shared on its behalf, and after the transaction takes place, transmits the data to the data consumer.

3.2. High-Level Overview. As shown in Figure 3, we presented a high-level overview of the system as follows:

Step 0: Sensor owners or data consumers who want to join in the system should complete the registration procedure with the service provider at first.

Step 1: A sensor owner uses a wireless connection with the service provider for data transmission, where it needs to transmit the sales information and the AES-encrypted sensor data to the service provider.

Step 2: After the service provider reviewed the sales information and received the encrypted data, it publishes the sales information on the blockchain platform.

Step 3: A data consumer selects a target seller based on its own interests and the sensor owner's reputation. After that, it needs to upload its request information (e.g., public key and pseudonym) to the smart contract and pay the deposit.

Step 4: If a sensor owner agrees to sell sensing data to a data consumer, it first accepts the data consumer's request and later encrypts the data key by employing a signcryption algorithm, and finally uploads it to the smart contract.

Step 5: The data consumer downloads the corresponding file from the platform, decrypts it to obtain the data key, where the balance is automatically deducted from the data consumer's account.

Step 6: The service provider transmits the encrypted data to the data consumer, and the data consumer uses the key to decrypt the encrypted data. If the data consumer finds that the key is invalid, it may submit an appeal to the service provider.

3.3. Design Goals and Security Requirements. The following are the design goals and security requirements that is considered in our proposed system.

(1) Privacy-preserving. Sensor owners and data consumers should have a certain number of pseudonyms in the system that they use to use sensor services, and their real identities should be hidden.

(2) Unlinkability and revocability. All pseudonyms registered on the service platform by sensor owners and consumers using their real identities cannot be connected. But when users seriously violate the rules, the system should have the right to reveal the real identity behind the pseudonym and revoke their right to use the service.

(3) Data integrity and reliability. When the sensor owner encrypts and sends the data to the service provider, the service provider does not have the decryption key, so it only temporarily stores the data and cannot tamper or delete the data without permission. Therefore, the integrity and reliability of the data are guaranteed.

(4) Fairness. On one hand, data consumers cannot obtain sensor data without paying a corresponding deposit. On the other hand, sensor owners should be caught and penalized if they provide invalid sensor data to sensor data consumers.

4. System Design

For our proposed sensor data sharing system in microgrid, we gave a formal description of the system running flow.

4.1. Running Flow. The running flow of the system consists of initialization, registration, publication, request, response, retrieval, guarantee, and evaluation phase. In addition, the

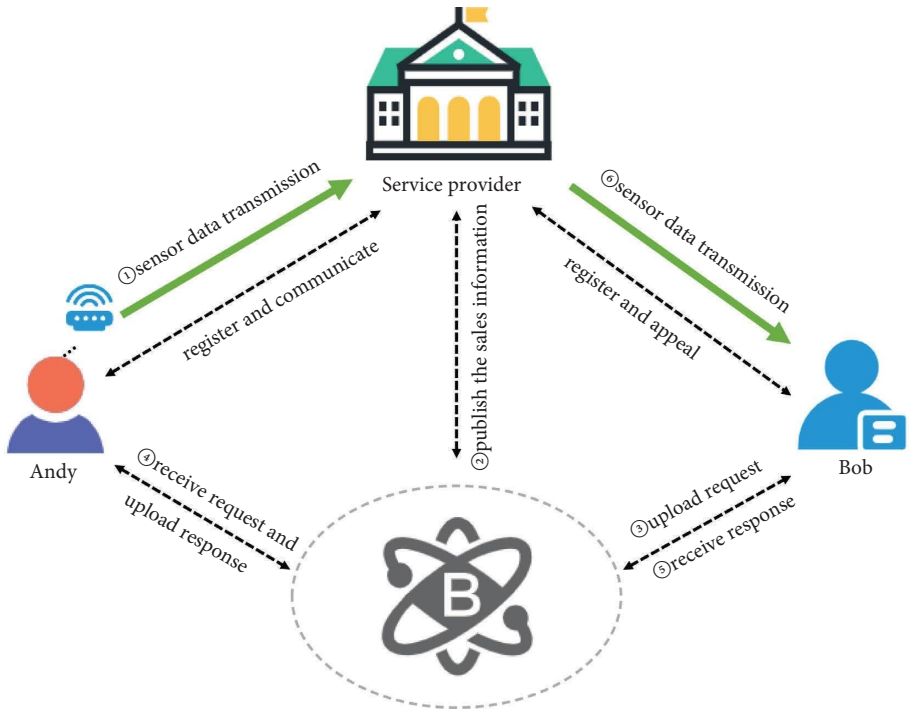


Figure 3: System instantiation scene.

functions in the smart contract includes uploadRegInfo, publishSales, uploadRequest, getRequest, uploadRes, getRes, submitAppeal, and calReputation. In particular, these functions are mainly focused on uploading registration information, publishing sales information, uploading purchase requests, receiving purchase requests, uploading responses information, receiving responses, submitting appeals, and calculating reputation in the above process. In particular, the formal descriptions of the proposed system are as follows:

4.1.1. Initialization. The users (sensor owners and data consumers) are authenticated by the ABS algorithm before they participate in the system. Specifically, the system calls ABS.Setup() to set public parameter PP and master key MK, and sends the access structure T to the user, while the user needs to submit its attribute set γ_u that satisfies T. Then, ABS.KeyGen() inputs MK, PP, and γ to generate the user's private key SK_γ. Finally, the algorithm ABS.Sign() outputs a signature σ_u that satisfies the condition.

4.1.2. Registration. The service provider in the system provides registration service. Both the sensor owner and the sensor data consumer must complete the registration in the service provider. Only the registered entities can enjoy the services in the system. First, the registration entity submits the corresponding information (γ_u, rid, and σ_u) to the service provider(rid is the real identity of the registered user). Then, the service provider calls the ABS.verify() to verify the authenticity of the registration information. Only those who have passed the verification can complete the registration, otherwise the registration will fail. Then, the system automatically generates a certain number of pseudonyms pid_u based on the attribute set of the registered user to protect their privacy. Finally, γ_u and pid_u will be submitted to the blockchain through uploadRegInfo. The registration and uploadReginfo algorithm is described in Algorithm 1. Note that the service provider locally stores the real identity of the registered user for subsequent tracking of requirements.

4.1.3. Publication. When a sensor owner wants to publish data sales information, he first needs to call the AES.enc(k, m_d) to encrypt the data into ciphertext c_d, and then transmit it to the service provider through the sensor (each sensor has a corresponding ID sid). The service provider receives the ciphertext and publishes the sales information through publishSales in the smart contract (as shown in Algorithm 2). The published information includes the seller's pseudonym pid_i, the information info of the sensor data info, and the expected price p.

4.1.4. Request. Sensor consumers can choose the data they are interested in or want to buy based on the published sales information. If the sensor consumer selects a certain sensor owner, that is, the seller, upload the consumer's own public key P_b, its own pseudonym pid_j, the seller's pseudonym pid_i, and the index of the data index in the hyper elliptic curve-based signcryption algorithm to the smart contract through uploadRequest in Algorithm 3. As for how consumers can choose sellers efficiently, they can make decisions based on the reputation value of the sellers, which will be described in detail in Section 4.2.

Require: a user's attribute set γ_u and real identity r_{id}.
Ensure: the pseudonym of users p_{idu}.
(1) **if** ABS.verify($b = 1$) **then**
(2) register successful;
(3) uploadReginfo $(\gamma_u, \mathrm{pid}_u)$;
(4) **else**
(5) return false;
(6) **end if**

Algorithm 1: Register and uploadReginfo.

Require: service provider has received the sensor data.
Ensure: publish sales information successfully.
(1) **if** RS (receive status) = true; **then**
(2) Sales.sid = sid;
(3) Sales.pidu = pidu;
(4) Sales.info = info;
(5) Sales.price = p;
(6) **else**
(7) return false;
(8) **end if**

Algorithm 2: publishSales (sid, pidu, info, p).

Require: Consumer's public key, pseudonyms of both parties, and data index
Ensure: upload request successfully.
(1) **if** SS (selected status) = true; **then**
(2) Req.cpk = P_b;
(3) Req.consumer = pid_j;
(4) Req.owner = pid_i;
(5) Req.data = index;
(6) **end if**

Algorithm 3: uploadRequest $(P_b, \mathrm{pid}_j, \mathrm{pid}_i, \mathrm{index})$.

4.1.5. Response. The sensor owner can obtain the consumer's request information through getRequest in the smart contract. If the owner agrees with the quotation and other matters in the request information, it encrypts its own symmetric key through signcryption (r, d_a, m_k, P_b, and P_a) and obtains the transmission text (c_k, R, s). Here, c_k refers to the ciphertext of the sensor owner in the signcryption algorithm, while R and s are the generated signature; d_a refers to the private key of the sensor owner, P_a and P_b are the public keys of the sensor owner and the data consumer. Then, it sends a tip information tip that agrees to the request to the service provider and calls uploadRes as in Algorithm 4 to upload the transmitted text to the smart contract.

4.1.6. Retrieval. The data consumer gets the corresponding file from getRes in the smart contract, and decrypts it through unsigncryption $(P_b, P_a, d_b, h, c_k, R, s)$ to obtain the data key k. Note that the check() algorithm in unsigncryption can verify whether the signcryption calculation is performed using the public key provided by the consumer, which provides verifiability. After that, the consumer requests the encrypted data from the service provider. Of course, for the security of the transaction, the service provider will verify that the sensor owner has agreed to the request before transmitting the data to the consumer. Finally, when the consumer receives the data transmitted by the provider, he decrypts the original sensor data m_d using the AES.dec (k, c_d). Note that when sensor data is obtained, the system will automatically debit the consumer's account and credit the remaining fee to the sensor owner's account.

4.1.7. Guarantee. If the consumer finds that the key is invalid, i.e., the sensor owner has provided a fake data key, he can submit an appeal to the service provider via submitAppeal. The service provider reverifies the situation and orders the sensor owner to provide the consumer with a valid key. If the sensor owner continues to provide invalid keys, the service provider will reveal the real identity rid

Require: send a tip to the service provider
Ensure: upload response successfully.
(1) **if** ST(status of tip) = true; **then**//the tip has been sent
(2) Res.text[] = c_k, R, s;
(3) Res.owner = pid_i;
(4) Res.consumer = pid_j;
(5) **else**
(6) return "please send a tip agreeing to the request";
(7) **end if**

Algorithm 4: uploadRes $(c_k, R, s, \text{pi}d_i, \text{pi}d_j)$.

behind the pseudonym and block all pseudonyms pidu, and also the data consumer's funds will be returned.

4.1.8. Evaluation. After a transaction cycle is completed, the data consumer can evaluate the transaction, that is, score and evaluate the sensor owner. The range of evaluation points is set from 1 to 10 points, with 6 points or more being positive evaluations and the following being negative evaluations. The evaluation within 1 month is the recent evaluation, otherwise it is the past evaluation, and the time window is 6 months. The calReputation in the smart contract automatically calculates the reputation of the sensor owner SR based on the above factors.

4.2. Reputation Calculation Model. In the proposed data sharing system, the reputation value of a sensor owner (seller) can be computed in real time by using the reputation calculation model, where the seller with a high reputation means that their data quality and transaction reputation are relatively good. Therefore, data consumers (buyers) can selectively choose sellers with high reputation for data trading. The reputation of the sensor owner is mainly affected by two key factors, transaction frequency, and after-sales evaluation. We combine these two factors to build a reputation calculation model to help consumers choose the right sellers efficiently.

(i) Transaction Frequency: The transaction frequency refers to the ratio of the number of transactions between sensor owner i and data consumer j to the average number of transactions between sensor owner i and other data consumers within the time window T, namely,

$$TF_{i\longrightarrow j} = \frac{N_{i\longrightarrow j}}{\overline{N_i}}, \tag{1}$$

where $N_{i\longrightarrow j} = (\alpha_i + \beta_i)$ and $\overline{N_i} = 1/|M|\sum_{m\in M} N_{i\longrightarrow m}$ (M is the total number of data consumers m transacting with sensor owner i within a time window). In conclusion, higher transaction frequency indirectly indicates a higher reputation of the sensor owner.

(ii) Evaluation Timeliness: Data consumers can rate sellers within one month after the transaction. In order to calculate reputation more accurately, the system assumes that recent reviews have a greater impact on the seller's reputation, while past reviews have less impact. Also, negative reviews have a greater impact on sellers than positive reviews. Therefore, we set the weight of recent evaluations to be ζ, the weight of past evaluations to be σ ($\zeta + \sigma = 1$, $\zeta > \sigma$), and the recent and past time periods to be one month. Positive reviews are weighted θ and negative reviews are weighted τ ($\theta + \tau = 1$ and $\theta < \tau$). Taking into account two sets of factors, the update transaction frequency formula is as follows:

$$\begin{cases} \alpha_i &= \zeta\theta\alpha_1^i + \sigma\theta\alpha_2^i, \\ \beta_i &= \zeta\tau\beta_1^i + \sigma\tau\beta_2^i. \end{cases} \tag{2}$$

Among them, when the current time t satisfies $t \le 1$ (month), the number of recent positive evaluations is α_1^i, and the number of recent negative evaluations is β_1^i. For $t > 1$, the number of positive and negative past events are α_2^i and β_2^i, respectively. Therefore, the reputation calculation function of the data seller (SR) is as follows:

$$\text{SR} = \frac{N_{i\longrightarrow j}}{\overline{N_i}} = \frac{\theta\left(\zeta\alpha_1^i + \sigma\alpha_2^i\right) + \tau\left(\zeta\beta_1^i + \sigma\beta_2^i\right)}{1/|M|\sum_{m\in M} N_{i\longrightarrow m}}. \tag{3}$$

In summary, the calReputation in the smart contract will automatically calculate and present the reputation of the sensor owner in real time according to the function. Data consumers can choose data sellers with high reputation for transactions. Of course, the price of data from sellers with high reputation will be higher.

5. Security Analysis

In this section, we present security analysis of our proposed system.

5.1. Privacy Preserving. The system adopts the ABS signature algorithm in the entity registration stage, and ABS has anonymity. Second, after the user is registered with the service provider, a certain number of pseudonyms pidus are returned for them to use when transacting. These hide the user's real identity and protect the user's privacy well.

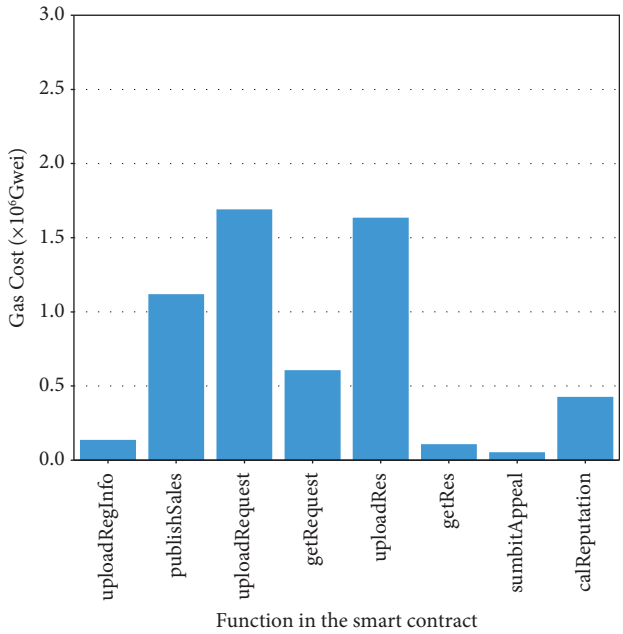


Figure 4: Gas cost of the function in smart contract.

5.2. Unlinkability and Revocability. The pseudonyms given by the service provider to the registered entity are only related to the real identity behind it, and there is no link between the pseudonyms. Since the service provider stores the real identity of the user locally, when the user acts dishonestly, the service provider will revoke the right to use the service under a pseudonym.

5.3. Data Integrity and Reliability. The sensor owner encrypts the data in the sensor with the AES algorithm and uploads it to the service provider, and the service provider stores it on their behalf. In the process of data transmission and storage, if there is no corresponding data key k, no one can modify and read the data.

5.4. Fairness. There is no third-party intervention in our system during the data transaction process. The service provider only provides the functions of registration, data storage and transmission, and does not enter into the process of data transaction. Additionally, data consumers can submit appeals when sensor owners provide invalid keys. These protect the rights and interests of consumers and ensure the fairness of the system.

6. Experimental Study

In this section, we evaluate the performance of the system, including testing out the gas cost of smart contracts and calculating the computational and communication cost of the cryptographic algorithms used. In particular, to facilitate compiling and testing smart contracts, we implement preops on Remix, a browser-based integrated development environment (IDE) for Ethereum. Specifically, the specific configuration in Remix includes the following: programming language (Solidity), compiler version (>=0.4.22 <0.7.0), and EVM version (default setting). In addition, we also evaluated the communication overhead and the computational cost of specific algorithms at each stage in the system running process.

Table 1: Computation cost of ECPM and HECDM.

Notation	Computation cost (ms)
ECPM	4.24
HECDM	2.2

6.1. Performance of Smart Contracts. The smart contract in the system consists of eight main functions, namely uploadRegInfo, publishSales, uploadRequest, getRequest, uploadRes, getRes, submitAppeal, and calReputation. The total gas cost of deploying smart contracts in the system is 1.0186×10^7Gwei, and the gas cost of each part of the function is 0.133, 1.114, 1.686, 0.602, 1.631, 0.103, 0.049, and 0.422 ($\times 10^6$Gwei), as shown in Figure 4. Among them, the reputation calculation for the sensor owner consumes a lot of gas, but it achieves our expected effect.

6.2. Communication and Computational Cost. We evaluated the communication and time cost of our system based on the benchmark given by [11], where the hardware is configured as a computer running jdk1.6, with 2 Intel CPU cores, a processing speed of 2.00 GHz, and a main memory capacity of 4 GB. As measured in [11], Table 1 shows time cost for elliptic curve point multiplication and hyperelliptic curve divisor-scalar multiplication, where a single scalar multiplication operation is respective 4.24 ms and 2.2 ms, and we use [12]'s ABS scheme to instantiate our system. In particular, we list some basic symbols in system cryptographic algorithms in Table 2 along with their cost. Therefore, we used these notations to calculate theoretical communication cost for different stages of the system's operational flow. As shown in Table 3, we only consider dominant operations for calculation, and the communication cost corresponding to initialization, publishing, request, response, and retrieval are 3040 bytes, 40 bytes, 65 bytes, 26 bytes, 66 bytes, and 72 bytes, respectively. In addition, the computational cost of the substeps of the ABS where the number of attributes is 50 and HECCS algorithms in the system are given in Table 4, which are 216.24 ms, 216.24 ms, 220.48 ms, 6.6 ms, and 4.4 ms, respectively.

6.3. Reputation Calculation Analysis. Since the reputation of the sensor owner is affected by the real-time evaluation and the positive and negative effects, as shown in Figure 5, we test the changes of the reputation value in the two time periods of $0 \sim 1$ month and $1 \sim 6$ months, respectively. Specifically, we preset $\theta = 0.3$, $\tau = 0.7$, $\zeta = 0.6$, and $\sigma = 0.4$ in the program, and take half a year as a time window. It can be clearly seen from Figure 5 that the greater the proportion of negative reviews within a month, the faster the decline in reputation value. On the other hand, the number of negative reviews

Table 2: Notation, definition, and size.

Notation	Definition	Size (byte)
$\|\mathbb{G}\|$	Size of an element in $\mathbb{G}$	20
$\|Z_p^*\|$	Size of an element of a group Z_p^*	20
$\|\sigma_u\|$	Size of a ABS signature	40
$\|K_h\|$	Size of a HECCS session key	16
$\|\sigma_h\|$	Size of a HECCS signature	56
|sid|	Size of a sensor identity	4
$\|pid_u\|$	Size of a pseudonym	5
|info|	Size of the data information	40
$\|c_{AES}\|$	Size of a AES ciphertext	16

Table 3: Communication cost of each phase.

Phase	Communication cost (byte)
Initialization	3040
Registration	40
Publication	65
Request	26
Response	66
Retrieval	72

Table 4: Computation cost of each algorithm step.

Algorithm	Computation cost (ms)
ABS.Setup	216.24
ABS.KeyGen	—
ABS.Sign	216.24
ABS.Verify	220.48
Signcryption	6.6
Unsigncryption	4.4

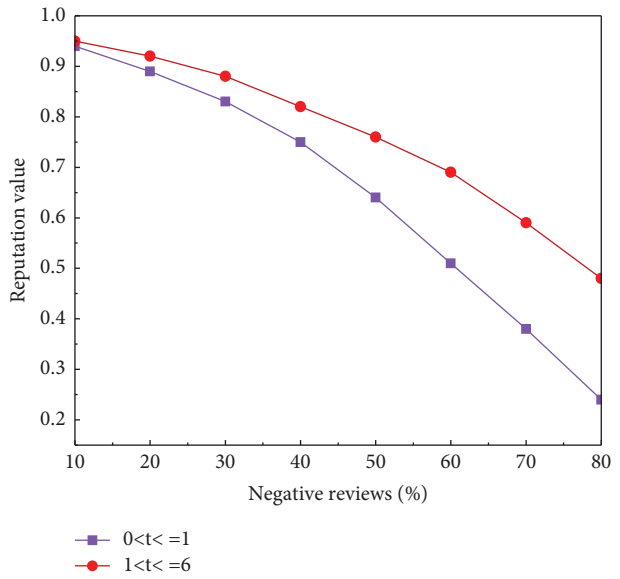


Figure 5: Reputation changes with different proportions of negative reviews.

between one month and six months has a lower impact on reputation value. This fully reflects the influence of the number of negative reviews and recent reviews on reputation value.

7. Related Work

Access control techniques are widely applied to share data from IoT sensors. Traditional access control techniques include discretionary access control (DAC) [13], role-based access control (RBAC) [14, 15], and capability-based access control (CapBAC) [16]. However, for these traditional models, a centralized authority is usually necessary to configure access control policies, resulting in centralized decision-making. Moreover, access control policies or records stored by a central third-party may be maliciously tampered with, leading to unreliable auditing. Facing this challenge, many attribute-based access controls [17] and attribute-based proxy re-encryption schemes [18, 19] have been proposed, but the issue of unreliable audits still exists in almost all of them. To settle the above issues, researchers combine blockchain technology with access control, which has the benefits of verifiability and decentralization.

Blockchain-based data sharing schemes have been presented in previous researches. Regarding data sharing between individuals and others, Chowdhury et al. [20] proposed a data sharing architecture of personal data with a notarization service offered by blockchain, and applied a blockchain-based mechanism to protect the privacy and integrity of transaction data. For data collected by IoT sensors, Manzoor et al. [21] combined the blockchain technology with the proxy re-encryption scheme to address the third-party trust issues of traditional IoT data sharing and improve scalability while guaranteeing data security. Since there are significant security issues in sharing data among users in multiple organizations, amounts of research has been conducted recently. Chen et al. [6] presented a blockchain-based privacy protection scheme based on k-anonymity and searchable, which achieves security and privacy protection of data in data sharing systems. However, the scheme requires further optimization and improvement for multiple groups data. Based on the Ethereum blockchain technology, Song et al. [22] accomplished the decentralization of the big data sharing system. However, these schemes mainly address data security and privacy issues and fail to focus on improving fairness in data sharing. To achieve anonymity and traceability of users, Huang et al. [7] utilized group signature technology in the proposed data sharing scheme without a trusted auditor by virtue of blockchain technology. Blockchain-based data sharing solutions are not only proposed in theory, but also play a significant role in solving difficulties in the life. To address the security and privacy concerns posed by electronic medical records, Chen et al. [23] proposed a signature based on antiquantum properties to share data securely with the blockchain. Tan et al. [24] proposed a blockchain-empowered solution that allows for direct tracking and revocation of medical records. To protect data privacy in building information model data sharing, Wang et al. [25]

proposed a blockchain-based approach, which can used to secure information in the next generation of smart building industrial IoT. These schemes motivated the achieved property of user traceability and revocability in our proposed data sharing scheme.

To further enhance fairness in the data sharing process, a number of blockchain-based solutions and architectures [26–28] have been proposed to ensure the security and fairness while implementing outsourcing services. Furthermore, Samuel et al. [8] presented a reputation system, fairly compensating through blockchain and differential privacy. In order to enhance the verifiability and fairness of cloud data management, Ge et al. [29] introduced a novel attribute-based proxy re-encryption scheme, according to which a concept called VA-ABPRE is defined and a concrete scheme is conducted. However, these schemes are constructed in the field of data outsourcing services while they cannot be directly deployed in microgrid. Wang et al. [9] applied blockchain technology to a supply chain to address issues such as distrust and asymmetric valuation of data that can arise from data sharing between upstream and downstream entities in the supply chain. But this study proposal uses an idealized model; actual supply chains cannot be completely adapted to it. Zhang et al. [30] introduced a data sharing scheme based on blockchain and ciphertext policy attribute-based encryption, where fair retrieval of ciphertexts is achieved through smart contracts. The editable blockchain in the authentication scheme of Zhai et al. [31] provided fine-grained and fair checksum functionality. Damisa et al. [32] proposed an Ethereum smart contract using a double auction mechanism to drive fairness and transparency in selection and compensation. The implementation of these smart contracts has effectively reduced the cost of manual intervention, where the property of accountability is not well studied [33].

8. Conclusions

In this work, we proposed a lightweight and privacy-preserving sensor data sharing system with attribute-based authorization in microgrid. In the system, we combined blockchain, smart contracts, and cryptographic algorithms (e.g., ABS, AES, and a lightweight signcryption scheme) to construct such sensor data sharing platform. Finally, we conducted a couple of experimental to evaluate the gas cost of functions in the smart contracts and general computational cost of cryptographic algorithms. To further improve the fairness of data sharing and running performance of the system, we continued to investigate the data pricing and claims mechanism, and moreover design a more lightweight cryptographic algorithm to replace the currently employed ABS algorithm.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was sponsored by National Natural Science Foundation of China (U1936213), Shanghai Rising-Star Program (no. 22QA1403800), Shanghai Sailing Program (no. 21YF1415000), and Program of Shanghai Academic Research Leader (no. 21XD1421500).

References

[1] A. Muhtadi, D. Pandit, N. Nguyen, and J. Mitra, "Distributed energy resources based microgrid: review of architecture, control, and reliability," *IEEE Transactions on Industry Applications*, vol. 57, no. 3, pp. 2223–2235, 2021.

[2] Y. E. Oktian, E. N. Witanto, and S. G. Lee, "A conceptual architecture in decentralizing computing, storage, and networking aspect of iot infrastructure," *IoT*, vol. 2, no. 2, pp. 205–221, 2021.

[3] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by internet of things," *Transactions on emerging telecommunications technologies*, vol. 25, no. 1, pp. 81–93, 2014.

[4] C. Lin, D. He, S. Zeadally, X. Huang, and Z. Liu, "Blockchain-based data sharing system for sensing-as-a-service in smart cities," *ACM Transactions on Internet Technology*, vol. 21, no. 2, pp. 1–21, 2021.

[5] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, vol. 15, p. 21260, 2008.

[6] Y. Chen, L. Meng, H. Zhou, and G. Xue, "A blockchain-based medical data sharing mechanism with attribute-based access control and privacy protection," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6685762, 12 pages, 2021.

[7] H. Huang, X. Chen, and J. Wang, "Blockchain-based multiple groups data sharing with anonymity and traceability," *Science China Information Sciences*, vol. 63, no. 3, pp. 130101–130113, 2020.

[8] O. Samuel, N. Javaid, M. Awais, Z. Ahmed, M. Imran, and M. Guizani, "A blockchain model for fair data sharing in deregulated smart grids," in *Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, IEEE, Waikoloa, HI, USA, December 2019.

[9] Z. Wang, Z. E. Zheng, W. Jiang, and S. Tang, "Blockchain-enabled data sharing in supply chains: model, operationalization, and tutorial," *Production and Operations Management*, vol. 30, no. 7, pp. 1965–1985, 2021.

[10] L. Jin, M. H. Au, W. Susilo, D. Xie, and K. Ren, "Attribute-based signature and its applications," in *Proceedings of the 5th ACM symposium on information, computer and communications security*, pp. 60–69, Kyoto, Japan, June 2010.

[11] S. A. Ch, N. uddin, M. Sher, A. Ghani, H. Naqvi, and A. Irshad, "An efficient signcryption scheme with forward secrecy and public verifiability based on hyper elliptic curve cryptography," *Multimedia Tools and Applications*, vol. 74, no. 5, pp. 1711–1723, 2015.

[12] R. Ma and L. Du, "Attribute-based blind signature scheme based on elliptic curve cryptography," *IEEE Access*, vol. 10, pp. 34221–34227, 2022.

[13] J. Zamite, D. Domingos, M. J. Silva, and C. Santos, "Group-based discretionary access control in health related repositories," *Journal of Information Technology Research*, vol. 7, no. 1, pp. 78–94, 2014.

[14] A. Alshehri and R. Sandhu, "Access control models for virtual object communication in cloud-enabled iot," in *Proceedings of*

the 2017 IEEE international conference on information reuse and integration (IRI), pp. 16–25, IEEE, San Diego, CA, USA, August 2017.

[15] M. L. Damiani, E. Bertino, B. Catania, and P. Perlasca, "Geo-rbac: a spatially aware rbac," *ACM Transactions on Information and System Security*, vol. 10, no. 1, p. 2, 2007.

[16] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1189–1205, 2013.

[17] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pp. 523–528, Hangzhou, China, May 2013.

[18] K. Liang, L. Fang, W. Susilo, and S. Duncan, "A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security," in *Proceedings of the 2013 5th International Conference on Intelligent Networking and Collaborative Systems*, pp. 552–559, IEEE, Sanda-Shi, Japan, Dcember 2013.

[19] Y. Zhang, J. Li, X. Chen, and H. Li, "Anonymous attribute-based proxy re-encryption for access control in cloud computing," *Security and Communication Networks*, vol. 9, no. 14, pp. 2397–2411, 2016.

[20] M. J. M. Chowdhury, A. Colman, M. A. Kabir, J. Han, and S. Paul, "Blockchain as a notarization service for data sharing with personal data store," in *Proceedings of the 2018 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. 1330–1335, IEEE, New York, NY, USA, August 2018.

[21] A. Manzoor, M. Liyanage, B. An, S. S. Kanhere, and M. Ylianttila, "Blockchain based proxy re-encryption scheme for secure iot data sharing," in *Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 99–103, IEEE, Seoul, Korea (South), May 2019.

[22] S. Song, "An effective big data sharing prototype based on ethereum blockchain," *Scientific Programming*, vol. 202214 pages, 2022.

[23] X. Chen, S. Xu, T. Qin, Y. Cui, S. Gao, and W. Kong, "Aq–abs: anti-quantum attribute-based signature for emrs sharing with blockchain," in *Proceedings of the 2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1176–1181, IEEE, Austin, TX, USA, April 2022.

[24] L. Tan, K. Yu, N. Shi, C. Yang, W. Wei, and H. Lu, "Towards secure and privacy-preserving data sharing for covid-19 medical records: a blockchain-empowered approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 271–281, 2022.

[25] H. Wang, X. Hao, L. Yin, P. Gong, F. Xiong, and W. Ren, "A blockchain-based and privacy-protected method for sharing of bim big data," in *Proceedings of the 2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pp. 185–191, IEEE, Chengdu, China, April 2022.

[26] Y. Guan, H. Zheng, J. Shao, R. Lu, and G. Wei, "Fair outsourcing polynomial computation based on the blockchain," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2795–2808, 2022.

[27] H. Zhang, P. Gao, Y. Jia, J. Lin, and N. N. Xiong, "Machine learning on cloud with blockchain: a secure, verifiable and fair approach to outsource the linear regression," *IEEE Transactions on Network Science and Engineering*, vol. 9, 2021.

[28] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Outsourcing service fair payment based on blockchain and its applications in cloud computing," *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 1152–1166, 2021.

[29] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 2907–2919, 2022.

[30] L. Zhang, T. Zhang, Q. Wu, Y. Mu, and F. Rezaeibagha, "Secure decentralized attribute-based sharing of personal health records with blockchain," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12482–12496, 2022.

[31] M. Zhai, Y. Ren, G. Feng, and X. Zhang, "Fine-grained and fair identity authentication scheme for mobile networks based on blockchain," *China Communications*, vol. 19, no. 6, pp. 35–49, 2022.

[32] U. Damisa and N. I. Nwulu, "Blockchain-based auctioning for energy storage sharing in a smart community," *Energies*, vol. 15, no. 6, p. 1954, 2022.

[33] REMIX, "remix ide for smart contract deployment provided by ethereum," 2023, https://remix.ethereum.org/.

Hindawi
Security and Communication Networks
Volume 2023, Article ID 7010155, 12 pages
https://doi.org/10.1155/2023/7010155

Research Article

Privacy-Preserving Industrial Control System Anomaly Detection Platform

Shan Gao,[1] Junjie Chen ,[1,2] Bingsheng Zhang,[1] and Kui Ren[1,3]

[1]*School of Cyber Science and Technology, Zhejiang University, Hangzhou, China*
[2]*Hangzhou Global Scientific and Technological Innovation Center, Zhejiang University, Hangzhou, China*
[3]*Zhejiang Provincial Key Laboratory of Blockchain and Cyberspace Governance, Hangzhou, China*

Correspondence should be addressed to Junjie Chen; 22121095@zju.edu.cn

Received 26 May 2022; Revised 9 July 2022; Accepted 24 November 2022; Published 28 April 2023

Academic Editor: Clemente Galdi

With the development of IT technologies, an increasing number of industrial control systems (ICSs) can be accessed from the public Internet (with authentication). In such an open environment, cyberattacks become a serious threat to both ICS system integrity and data privacy. As a countermeasure, anomaly detection systems are often deployed to analyze the network traffic. However, due to privacy regulation, the network packages cannot be directly processed in plaintext in many countries. In this work, we present a privacy-preserving anomaly detection platform for ICS. The platform consists of three nodes running low-latency MPC protocols to evaluate the live network packages using decision trees on the fly with privacy assurance. Our benchmark result shows that the platform can process thousands of packages every ten seconds.

1. Introduction

A modern industrial control system (ICS) is a complex distributed system that consists of multiple field devices, e.g., sensors, actuators, and instrumentation, as well as some control/management systems. ICS is the interface of cyber-physical system (CPS), enabling humans to control operations and receive data from devices. In recent years, ICS has been widely used in many industrial scenarios, such as gas, water, and nuclear power systems, and the security of these systems is critical.

As shown in Figure 1, a typical architecture of an industrial control system has four layers. (i) The enterprise management layer offers business services and is often connected to public network, which may include the enterprise resource planning (ERP) system, manufacturing execution system (MES), and management information system (MIS). (ii) The supervisory control layer receives and stores data from the underlying devices and then gives appropriate responses. (iii) The process control layer has programmable logic controller (PLC) and remote terminal unit (RTU), which directly control devices in the underlying layer. (iv) The field control layer has multiple field devices that receive commands and send data to the process control layer. As the enterprise management layer connects to the public network, ICS is exposed to cyberattacks. Along with the advancement of cyberattacks, the corresponding countermeasure techniques also need to be upgraded. In practice, a great number of famous industrial control systems have been severely threatened by cyberattack. For instance, the Stuxnet virus spied and reprogrammed industrial systems controlling centrifuges of the Iran nuclear power plant [1]. In 2021, hackers breached Colonial Pipeline using compromised password and Colonial Pipeline had to give hackers ransom [2].

To enhance industrial control system security, defense systems like intrusion (or anomaly) detection system (IDS) are deployed in ICS. IDS plays an important role in protecting ICS, which is commonly used to detect potential cyberattacks. IDS can be classified as network intrusion detection system (NIDS) and host-based intrusion detection system (HIDS). The NIDS examines network traffic, while the HIDS monitors the system data logs. According to detection approach, IDS can be classified as signature-based

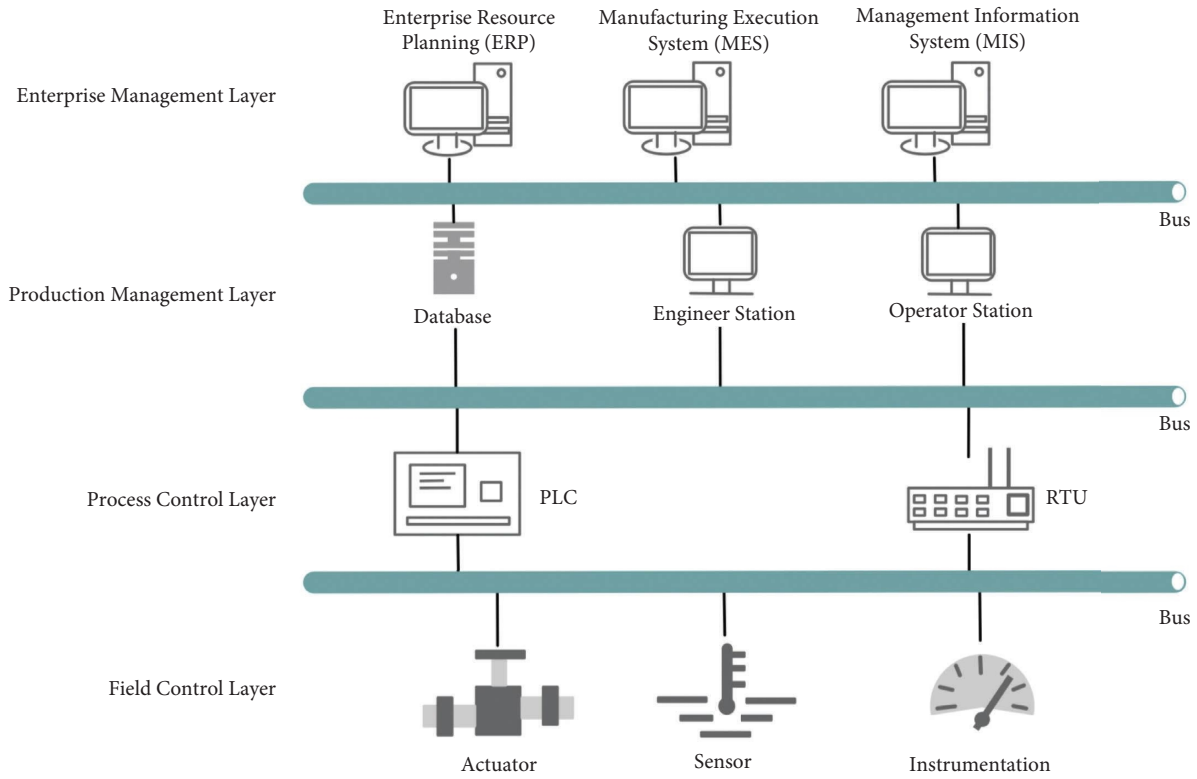


Figure 1: A typical architecture of industrial control system.

detection and anomaly-based detection. The former detects intrusion by recognizing harmful system pattern, while the latter does it by analyzing network traffic packages.

In this work, we aim to design an anomaly-based network intrusion detection platform for ICS. The platform can be deployed alongside any existing off-the-shelf ICSs, and it can examine live network packages on the fly and raise alarms once fault is detected. However, in many countries, processing network packages in plaintext violates the local privacy laws and regulations. The European Union has put forward *General Data Protection Regulation* [3] in 2016, which has a clear standard for the processing of personal information. In 2020, The United States carried out the *California Consumer Privacy Act* [4], creating a series of privacy rights for consumers, such as the right to access, delete, and know. In China, new 2020 edition of the *Personal Information Security Specification* [5] has proposed clear regulations in the life cycle of the personal information, including collection, storage, use, processing, transmission, openness, and deletion. These regulations will have a profound impact to systems that store and/or process personal information. Therefore, our anomaly detection platform is designed to be privacy preserving.

As a closely related work, Gao et al. [6] used the homomorphic encryption scheme to encrypt data when training and applying ICS-specific anomaly detection model. But homomorphic encryption scheme will lead to heavy computation overhead. Alternatively, we utilize low-latency secure multi-party computation (MPC) techniques for privacy-preserving anomaly detection.

More specifically, our platform consists of three non-colluding servers that run low-latency MPC protocols to analyze network package in real time using the gradient boosting decision tree (GBDT) model with privacy assurance. GBDT is an effective machine learning algorithm which classifies input data rapidly with high accuracy.

1.1. Our Contributions. In this work, we present an efficient MPC-based privacy-preserving anomaly detection platform for ICS. More specifically, the contributions of this work are as follows:

(i) We propose a new MPC-based anomaly detection architecture for ICS, and it is compatible with any off-the-shelf ICSs.

(ii) We design several new constant-round low-latency MPC protocols for privacy-preserving decision tree evaluation.

(iii) We implement a prototype of the proposed system, and our benchmark result shows that processing 1000 network packages with a depth-9 decision tree takes 11 seconds in the LAN setting.

1.2. Roadmap. The remainder of this paper is organized as follows. We introduce the preliminary knowledge about the approach we used in Section 2. Then, system overview and security model of our platform are given in Section 3. Section 4 describes privacy-preserving decision tree evaluation in detail. We present the performance of the proposed platform

in Section 5. The related work is given in Section 6. Finally, conclusion and future work are given in Section 7.

2. Preliminary

2.1. Notations. Throughout the paper, we use the following notations. Denote τ as the security parameter. Denote a value x indexed by a label i as $(x)_i$. (s, t)-secret sharing is to divide secret into t parts, and any s participants can reveal secret jointly. Denote (2, 2)-additive secret sharing, (3, 3)-additive secret sharing, and (2, 3)-additive secret sharing in $\mathbb{Z}_n$ in Table 1.

$r \longleftarrow R$ means to randomly sample the element r from the set R. In addition, $y \longleftarrow f(x)$ represents y is the output when the function $f(\,)$ takes x as input. For $x \in [-2^{\ell-1}, 2^{\ell-1}]$, map it to $\mathbb{Z}_{2^\ell}$ by adding $2^{\ell-1}$.

2.2. Gradient Boosting Decision Tree. Our proposed platform mainly uses gradient boosting decision tree (GBDT) as intrusion detection model. Decision tree is a classical machine learning model, which is efficient and interpretable. Its non-leaf node is decision node, which performs a test to decide to go to left sub-tree or right sub-tree. Its leaf node is the end of a decision path that begins with root node, including prediction result. Boosting is a kind of algorithm that combines many weak learners into a strong learner. The first step is training a base learner, like decision tree. Then, adjust training samples according to the classification result of base learner, so that those misclassified samples will get more attention in the subsequent training process. After that, train next weak learner using adjusted training samples. Repeat the process iteratively to obtain enough weak classifiers and combine them together according to their weight to obtain a strong classifier. Gradient boosting is an algorithm in boosting, which iterates the new learner through gradient descent.

The GBDT is a learning algorithm based on boosting. Its essence is that the next regression decision tree is built on the gradient descent direction of the loss function of the last round, and multiple regression decision trees are combined into a gradient boosting decision tree finally. When x is the input of GBDT, its classification result is $\widehat{y} = \sum_{k=1}^{K} f_k(x)$, where K is number of decision trees in GBDT and $f_k(x)$ is k-th tree's output. In general, the tree in GBDT is CART tree.

Given a training set S: $= \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i$ is input feature vector and y_i is its class label. The process of training GBDT consists of T rounds iteration. In the k-th iteration, the goal is to generate a decision tree f_k to minimize the objective function L.

$$L^{(k)} = \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(k-1)} + f_k(\mathbf{x}_i)\right) + \Omega(f_k), \tag{1}$$

where $\Omega(f) = \gamma T + 1/2\lambda\sum_{j=1}^{T}\omega_j^2$ is regularization item and $\widehat{y}_i^{(k-1)}$ is i-th sample's classification result in k-th iteration. l is loss function, T is number of leaf nodes, and ω_j is the value of a leaf node. Expression (1) uses a second-order Taylor expansion to get the following expression.

$$L^{(k)} \simeq \sum_{i=1}^{n}\left[l\left(y_i, \widehat{y}^{(k-1)}\right) + g_i f_k(\mathbf{x}_i) + \frac{1}{2}h_i f_k^2(\mathbf{x}_i)\right] + \Omega(f_k), \tag{2}$$

where $g_i = \partial_{\widehat{y}^{(k-1)}} l(y_i, \widehat{y}^{(k-1)})$ is the first step degree value of loss function l and $h_i = \partial^2_{\widehat{y}^{(k-1)}} l(y_i, \widehat{y}^{(k-1)})$ is the second step degree value of loss function l.

The k-th decision tree only includes a root node with the training set S initially. Suppose a sample set S in a node is partitioned into S_l and S_r; $L_{\text{split}}(S_l, S_r)$ is defined as follows.

$$L_{\text{split}}(S_l, S_r) = -\frac{1}{2}\left[\frac{\left(\sum_{i\in S_l} g_i\right)^2}{\sum_{i\in S_l} h_i + \lambda} + \frac{\left(\sum_{i\in S_r} g_i\right)^2}{\sum_{i\in S_r} h_i + \lambda}\right] + \gamma T. \tag{3}$$

Perform a test for each possible split point and select the optimal split point that causes minimum L_{split}. If the current node does not meet the splitting requirements, for example, the depth of the current node reaches the maximum, the current node becomes a leaf node with a value $V(S)$, and $V(S)$ is defined as follows.

$$V(S) = -\frac{\sum_{i\in S} g_i}{\sum_{i\in S} h_i + \lambda}. \tag{4}$$

GBDT has strong classification ability in anomaly detection task.

2.3. Secure Multi-Party Computation. Secure multi-party computation permits two or more participating parties to obtain output result by jointly computing over sensitive data from respective inputs. At the same time, the participating parties do not learn more about other parties' inputs than the information about the output, so that each participating party can get computation result without leaking sensitive message.

Secure multi-party computing usually includes two different adversary models, namely, semi-honest security model and malicious security model. A semi-honest security model is one in which the adversary will honestly perform the intended calculation process but may wish to know the information of each party to the maximum extent. A malicious security model is one in which an adversary can control, manipulate, and arbitrarily contaminate information on a multi-party computing network. In this work, we mainly consider the semi-honest security model.

Although the first MPC protocol was already proposed by A. C.-C. Yao [7] in the 1980s, it was implemented practically in the last eighteen years. Nowadays, MPC becomes more important as data privacy gets more and more attention. It was adopted for private set intersection [8] and privacy-preserving machine learning [9].

Table 1: (s,t)-secret sharing.

(s,t)-additive sharing in $\mathbb{Z}_n$	Description
(2, 2)-additive sharing	$[x] := \{(x)_0, (x)_1\}$, where $x = (x)_0 + (x)_1 \pmod n$. S_0 holds $(x)_0$ and S_1 holds $(x)_1$
(3, 3)-additive sharing	$\langle x\rangle := \{(x)_0, (x)_1, (x)_2\}$, where $x = (x)_0 + (x)_1 + (x)_2 \pmod n$. S_0 holds $(x)_0$, S_1 holds $(x)_1$, and S_2 holds $(x)_2$
(2, 3)-replicated sharing	$\langle x\rangle^r := \{(x)_0, (x)_1, (x)_2\}$, where $x = (x)_0 + (x)_1 + (x)_2 \pmod n$. S_0 holds $\{(x)_0, (x)_1\}$, S_1 holds $\{(x)_1, (x)_2\}$, and S_2 holds $\{(x)_2, (x)_0\}$

Secret sharing is one of important parts in MPC. The remaining part of this section introduces distributed interval containment function (DICF) and shared oblivious transfer that we adopt in our MPC protocol.

2.3.1. Function Secret Sharing. Function secret sharing (FSS) [10] can split a function $f: \mathbb{X} \longrightarrow \mathbb{Y}$ into $\{(f)_0: \mathbb{X} \longrightarrow \mathbb{Y}, (f)_1: \mathbb{X} \longrightarrow \mathbb{Y}\}$ and $f(x) = (f(x))_0 + (f(x))_1 \pmod{|\mathbb{Y}|}$ for each $x \in \mathbb{X}$, where $|\mathbb{Y}|$ denotes the number of element in $\mathbb{Y}$. Distributed point function (DPF) is a FSS scheme. For a point function $f_{\alpha,\beta}(x): \mathbb{X} \longrightarrow \mathbb{Y}$, the range $\mathbb{Y}$ has only one non-zero value $f_{\alpha,\beta}(\alpha) = \beta$. There are two algorithms in DPF:

(i) $\mathrm{Gen}(1^\lambda, f_{\alpha,\beta})$: It generates a pair of keys $((\mathscr{F})_0 (\mathscr{F})_1)$. Each key is the share of $f_{\alpha,\beta}$ without revealing α and β.

(ii) $\mathrm{Eval}(i, (\mathscr{F})_i, x)$: $\forall x \in \mathbb{X}$, it outputs $(\beta_x)_i$, such that $(\beta_x)_0 + (\beta_x)_1 = f_{\alpha,\beta}(x) \pmod{|\mathbb{Y}|}$.

Denote run Eval on all inputs by $\mathrm{EvalAll}(i, (\mathscr{F})_i)$.

DICF [11] is also a FSS scheme that can judge whether a secret input value is in a publicly known interval. Denote an interval containment function as the following equation.

$$\mathbf{F}_{p,q}(x) = \begin{cases} 1, & if\ x \in [p,q], \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

DICF uses offset interval containment function defined as follows.

$$\mathbf{F}_{p,q,r_{\text{in}},r_{\text{out}}}(x + r_{\text{in}}) = \mathbf{F}_{p,q}(x) + r_{\text{out}}, \tag{6}$$

where r_{in} and r_{out} are random offset values. Like DPF, DICF also consists of two algorithms.

(i) $\mathrm{Gen}(1^\lambda, \mathbf{F}_{p,q,r_{\text{in}},r_{\text{out}}})$: it generates $((\mathscr{F})_0, (\mathscr{F})_1)$, as p, q are publicly known and r_{in} and r_{out} are unrevealed.

(ii) $\mathrm{Eval}(i, (\mathscr{F})_i, x + r_{\text{in}})$: outputs a result $(\beta)_i$, so that $(\beta)_0 + (\beta)_1 - r_{\text{out}} = \mathbf{F}_{p,q}(x) \pmod{|\mathbb{Y}|}$.

2.3.2. Oblivious Transfer. Oblivious transfer (OT) [12] is an important basic block in many MPC protocols. In oblivious transfer protocol, a sender has multiple messages and only one of them will be selected by receiver. Which message is selected is oblivious to the sender and the receiver can only obtain the selected message.

Shared OT is a kind of OT scheme that is used to fetch value in the shared form without revealing the value. In our approach, we utilize a 3-party shared OT protocol. In this protocol, three participants S_0, S_1, S_2 share a data vector $\mathbf{x} = (x_0, x_1, x_2, \ldots, x_{n-1})$ and an index i ($i \in \mathbb{Z}_n$), as S_0 holds $\{(\mathbf{x})_0, (\mathbf{x})_1, (i)_0\}$, S_1 holds $\{(\mathbf{x})_1, (\mathbf{x})_2, (i)_1\}$, and S_2 holds $\{(\mathbf{x})_2, (\mathbf{x})_0, (i)_2\}$, where $(x)_j = ((x_0)_j, (x_1)_j, (x_2)_j, \ldots, ((x_{n-1})_j))$. Then, they can fetch x_i in the shared form without revealing i by jointly computing.

2.4. Intrusion Detection. The main goal of intrusion detection system [13] is to detect cyberattacks. Cyberattack is any type of offensive action against computer systems, computer networks, or personal computer. Damaging, exposing, modifying, disabling software or services, or stealing or accessing data from any computer without authorization is considered an attack on the computer and computer network. According to the attack mode, cyberattack can be divided into active attack and passive attack. An active attack attempts to destroy computer system, which includes denial of service (DoS), distributed denial of service (DDoS), and botnet, while a passive attack aims to learn information about network system like port scan attack.

DoS deliberately attacks flaws in the network protocol implementation or depletes the target's resources by brutal means, so that service or network cannot provide normal services. DDoS is a special form of denial of service attack based on DoS. It is a distributed and coordinated large-scale attack that may come from multiple attackers.

Botnet refers to the use of one or more means of transmission to infect a large number of hosts with bot program virus, so as to form a one-to-many control network between the controller and the infected host. The attacking process of port scanning attack is usually to remotely scan each port of the target computer, detect the services provided by different ports, and then record the response of the target computer to collect its information.

Generally, network anomaly detection requires the information about data packets, such as packet header characteristics, characteristics about TLS, and packet length.

3. System Framework and Security Model

This section gives the overview of our system framework firstly and describes the security model in Section 3.2.

3.1. System Framework. There are several components in the system framework, as depicted in Figure 2. The ICS preprocessed its packages firstly, including extracting features and secret sharing. For each data package, a feature vector is extracted from it. The feature vector contains the information about packet header and packet length. Then, the feature vector is divided into three parts using (2, 3)-additive secret sharing among S_0, S_1, S_2. Next, the parts of secret are

distributed to three servers, where a well-trained CART model is stored in the shared form using (2, 2)-additive secret sharing between S_0 and S_1. Finally, the three nodes jointly obtain a detection result based on CART model, running MPC protocols described in Section 4.

3.2. Security Model. In the process of anomaly detection, we cannot store and process sensitive data from ICS in plaintext, according to respective laws and regulations. In order to detect attacks in network traffic with privacy assurance, we adopt MPC to achieve our goals. Firstly, we assume that there is a component in the ICS that can extract feature vectors of its network packages. This component shall be trusted, and it will then secretly share the extracted features to the three MPC nodes of our platform. One out of the three MPC nodes can be semi-honestly corrupted by the adversary. The shared process result will be sent back to the system admin of ICS, who will recover the result and make further actions accordingly.

3.2.1. Security Requirements. As described above, our proposed platform should protect privacy of ICS data when examining the sensitive data. Besides, the platform should respond accurately and quickly so that ICS can identify anomalies in time. Thus, we define the following key security requirements.

(i) *Data Privacy.* Though we detect the sensitive data from ICS, the data will not be stored or processed in plaintext. Even if a MPC node is semi-honestly corrupted by the adversary, the data privacy can still be protected.

(ii) *Accuracy.* As the platform's task is anomaly detection, the accuracy of detection model should be as high as possible.

(iii) *On Time.* Our platform should respond ICS as fast as possible so that ICS can handle cyberattack timely.

4. Privacy-Preserving Decision Tree Evaluation

This section describes the MPC protocols utilized in our approach. Firstly, we describe 3-party shared OT in Section 4.1. Then, we give the whole detection process, including data preprocessing, tree model storage, and evaluation.

4.1. 3-Party Shared OT. Given a replicated shared data vector $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ and an additively shared index $i \in \mathbb{Z}_n$, three participants hold the shared form $\{(\mathbf{x})_0, (\mathbf{x})_1, (i)_0\}$, $\{(\mathbf{x})_1, (\mathbf{x})_2, (i)_1\}$, and $\{(\mathbf{x})_2, (\mathbf{x})_0, (i)_2\}$ respectively, that is similar to [14]. Then, the three participants can obtain x_i in shared form by running our 3-party shared OT protocol.

4.1.1. Intuition. Our protocol is mainly constructed on the basis of Paul et al. [14], whose main idea is that each participant serves as the generator of DPF scheme, while the other two participants serve as evaluators to get i-th value of vector $\mathbf{x}$ in the shared form. For instance, let S_0 be the DPF generator and S_1, S_2 be the DPF evaluators. Firstly, S_1 and S_2 randomly select $r_1, r_2 \longleftarrow \mathbb{Z}_n$, respectively. Then, S_1, S_2 exchange $r_1 - (i)_1, r_2 - (i)_2$ and send r_1, r_2 to S_0. After that, S_1, S_2 compute $\sigma := r_1 - (i)_1 + r_2 - (i)_2 \pmod n$ and S_0 computes $\theta := (i)_0 + r_1 + r_2 \pmod n$. It is easy to see that $\theta - \sigma = i \pmod n$. Next, S_0 generates a pair of DPF keys for point function $f_{\theta,1}(x)$ and sends keys to evaluators. Finally, S_1, S_2 run full domain evaluation to jointly obtain $\{[\beta_{0,\theta}], [\beta_{1,\theta}], \ldots, [\beta_{n-1,\theta}]\}$ ($[\beta_{k,\theta}]$ is 1 if $k = \theta$, and is 0 otherwise). Note that θ-th element in shifted vector is $[(x_i)_2]$, as $[(\mathbf{x})_2]$ is cyclic shifted to the right σ position. After all these steps, S_1, S_2 hold $[(x_i)_2]$ in shared form. Following similar steps, S_0, S_2 can jointly get $[(x_i)_0]$ and S_0, S_1 can jointly get $[(x_i)_1]$. In our 3-party shared OT protocol, we let the generator produce DPF keys of $f_{\mu,1}(x)$, where $\mu \longleftarrow \mathbb{Z}_n$ is randomly picked by generator. Then, the generator can produce DPF keys, which leads to less communication. Subsequently, all participants jointly compute and reveal $\langle\sigma\rangle := \langle i\rangle + \langle 0\rangle - \mu$ to evaluators. At the end, evaluators can get $[(x_i)_0], [(x_i)_1], [(x_i)_2]$ in the shared form.

4.1.2. Protocol Description. The 3-party shared OT is depicted in Protocol 1. Initially, for $j \in \mathbb{Z}_3$, S_j and S_{j+1} agree on a random seed $\varphi_j \in \{0,1\}^\tau$ as $j \in \mathbb{Z}_3$, and if index $(j+1)$ greater than 2, S_{j+1} is the brief form of $S_{j+1 \pmod 3}$. Note that as the index $j \in \mathbb{Z}_3$, we omit (mod 3) in the rest of this paper. Before each round of shared OT, for $j \in \mathbb{Z}_3$, node S_j generates DPF keys $((\mathscr{F}_{\mu_j})_0, ((\mathscr{F}_{\mu_j})_1)$. Then, S_j sends $(\mathscr{F}_{\mu_j})_0$ to S_{j+1} and $(\mathscr{F}_{\mu_j})_1$ to S_{j+2}. All participants generate some $\langle 0\rangle$ using random seeds and pseudo-random function PRF. They jointly compute and reveal $\langle\sigma_j\rangle := \langle i\rangle + \langle 0\rangle - \mu_j \pmod N$ to evaluators S_{j+1} and S_{j+2}. Next, evaluators S_{j+1}, S_{j+2} use DPF keys $((\mathscr{F}_{\mu_j})_0, ((\mathscr{F}_{\mu_i})_1)$ to get $\{[\beta_{0,\mu_j}], [\beta_{1,\mu_j}], \ldots, [\beta_{n-1,\mu_j}]\}$ by running EvalAll algorithm. Then, they jointly obtain

$$\left[(x_i)_{j+2}\right] = \sum_{k=0}^{n-1}\left(\left(x_{k+\sigma_j}\right)_{j+2} \bullet \left[\beta_{k,\mu_j}\right]\right) \left(\mathrm{mod} 2^{\ell}\right). \tag{7}$$

They can jointly get x_i as $x_i = (x_i)_0 + (x_i)_1 + (x_i)_2$. Lastly, participants rerandomize shares to ensure their uniform distribution.

4.2. Data Preprocessing. Before ICS transmits its network packages, the data need to be preprocessed in two steps. Firstly, ICS extracts a feature vector for each package so that decision tree can detect on package level. Then, it completes data desensitization. A feature vector $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ is shared as $\langle x\rangle^r = \{(\mathbf{x})_0, (\mathbf{x})_1, (\mathbf{x})_2\}$, where $(x)_j = ((x_0)_j, (x_1)_j, \ldots, (x_{n-1})_j)$ and $j \in \mathbb{Z}_3$. Then S_0 holds $\{(\mathbf{x})_0, (\mathbf{x})_1\}$, S_1 holds $\{(\mathbf{x})_1, (\mathbf{x})_2\}$, and S_1 holds $\{(\mathbf{x})_2, (\mathbf{x})_0\}$.

4.3. Storage of Tree Model. As we adopt a constant-round MPC protocol that needs a full binary tree and our trained model is just a binary tree, we will pad the binary tree as

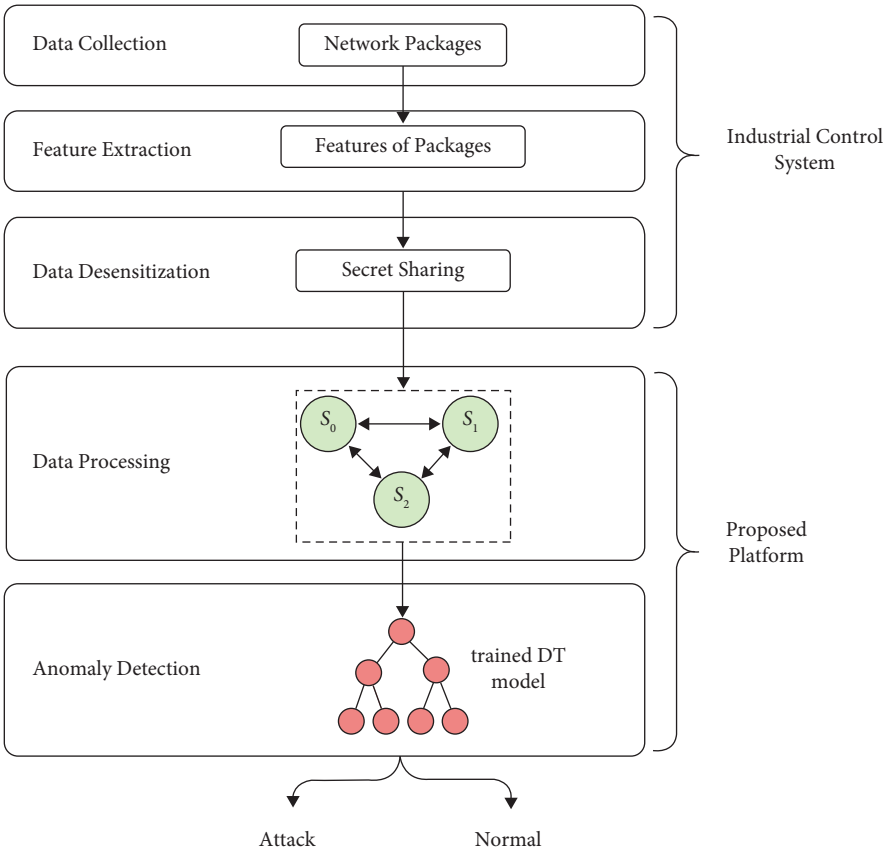


Figure 2: System framework.

Initialization:
  for *each* $j \in \mathbb{Z}_3$ **do**
    S_j and S_{j+1} have the same random seed $\varphi_j \longleftarrow \{0,1\}^\tau$;
  end
Preparing:
  for *each* S_j **do**
    Generate $\mu_j \longleftarrow \mathbb{Z}_n$;
    Generate a pair of keys $((\mathcal{F}_{\mu_j})_0, (\mathcal{F}_{\mu_j})_1)$ for $f_{\mu_j,1}$: $\mathbb{Z}_n \longrightarrow \mathbb{Z}_{2^\ell}$;
    Send $(\text{sid}, (\mathcal{F}_{\mu_j})_0)$ to S_{j+1}, $(\text{sid}, (\mathcal{F}_{\mu_j})_1)$ to S_{j+2};
  end
for *each* S_j **do**
  Receive $(\text{sid}, (x)_j, (x)_{j+1}, (i)_j)$ from the environment;
end
for *each* S_j **do**
  for *each* $k \in \mathbb{Z}_3$ **do**
    $r_{k,j} \longleftarrow \text{PRF}^{\mathbb{Z}_n}_{\varphi_j}(\text{sid}, k), r_{k,j+2} \longleftarrow \text{PRF}^{\mathbb{Z}_n}_{\varphi_{j+2}}(\text{sid}, k)$;
    $(\sigma_k)_j \longleftarrow (i)_j + r_{k,j} - r_{k,j+2} \pmod n$;
  end
  $(\sigma_j)_j \longleftarrow (\sigma_j)_j - \mu_j \pmod n$;
  Send $(\text{sid}, (\sigma_j)_j, (\sigma_{j+1})_j)$ to S_{j+2}, $(\text{sid}, (\sigma_j)_j, (\sigma_{j+2})_j)$ to S_{j+1};
end
for *each* S_j **do**
  Receive $(\text{sid}, (\sigma_{j+1})_{j+1}, ((\sigma_{j+2})_{j+1})$ from S_{j+1}, $(\text{sid}, (\sigma_{j+2})_{j+2}, ((\sigma_{j+1})_{j+2})$ from S_{j+2};
end
for *each* S_j **do**
  for *each* $k \in \mathbb{Z}_3$ **do**

Protocol 1: Continued.

$\sigma_k \longleftarrow (\sigma_k)_0 + (\sigma_k)_1 + (\sigma_k)_2$;
end
$\left\{(\beta_{0,\mu_{j+1}})_1, (\beta_{1,\mu_{j+1}})_1, \ldots, (\beta_{n-1,\mu_{j+1}})_1\right\} \longleftarrow$ DPF.EvalAll$(1, (\mathcal{F}_{\mu_{j+1}})_1)$;
$\left\{(\beta_{0,\mu_{j+2}})_0, (\beta_{1,\mu_{j+2}})_0, \ldots, (\beta_{n-1,\mu_{j+2}})_0\right\} \longleftarrow$ DPF.EvalAll$(0, (\mathcal{F}_{\mu_{j+2}})_0)$;
$(y)_j \longleftarrow \sum_{k=0}^{n-1}((x_{k+\sigma_{j+1}})_j \cdot (\beta_{k,\mu_{j+1}})_1 + (x_{k+\sigma_{j+2}})_{j+1} \cdot ((\beta_{k,\mu_{j+2}})_0$;
$\psi_j \longleftarrow \mathrm{PRF}_{\varphi_j}^{\mathbb{Z}_{2^\ell}}$(sid), $\psi_{j+2} \longleftarrow \mathrm{PRF}_{\varphi_{j+2}}^{\mathbb{Z}_{2^\ell}}$(sid);
Return $(y)_j := (y)_j + \psi_j - \psi_{j+2} \pmod{2^\ell}$;
end

Protocol 1: 3-party shared OT protocol.

depicted in Figure 3 when storing the tree model. Then, we get a full binary tree such that adding tree nodes does not affect the final result. This full binary tree can be saved as two vectors $\mathbf{N} = (N_0, N_1, \ldots, N_{\mathcal{N}-1})$ and $\mathbf{L} = (l_0, l_1, \ldots, l_{\mathcal{L}-1})$, where $\mathcal{N} := 2^{d-1} - 1$ is the number of non-leaf nodes and $\mathcal{L} := 2^{d-1}$ is the number of leaf nodes in a full binary tree with depth d. N_i ($N_i := \{t_i, v_i\}$) denotes the non-leaf node with index i. The index increases from top to bottom, left to right. If a non-leaf node has non-leaf sub-nodes, its left child node is N_{2i+1} and its right child node is N_{2i+2}. The values t_i and v_i belongs to the i-th non-leaf node. Given a feature vector, the decision tree algorithm extracts the t_i-th value of the feature vector to compare with the threshold v_i. If t_i-th value of feature vector is greater than v_i, perform the same operation on the right child node, otherwise on left child node. The algorithm will be end when the node is a leaf node. The l_i in $\mathbf{L}$ is classification result when the leaf node with index i is the end of decision path. $\mathbf{N}$ and $\mathbf{L}$ are shared as $[\mathbf{N}] = \{(\mathbf{N})_0, (\mathbf{N})_1\}$ and $[\mathbf{L}] = \{(\mathbf{L})_0, (\mathbf{L})_1\}$. Then, S_0 holds $\{(\mathbf{N})_0, (\mathbf{L})_0\}$ and S_1 holds $\{(\mathbf{N})_1, (\mathbf{L})_1\}$, where $(N)_j = (\{(t_0)_j, (v_0)_j\}, \{(t_1)_j, (v_1)_j\}, \ldots, \{(t_{\mathcal{N}-1})_j, (v_{\mathcal{N}-1})_j\})$, $(L)_j = ((l_0)_j, (l_1)_j, \ldots, (l_{\mathcal{L}-1})_j)$ and $j \in \mathbb{Z}_2$.

4.4. Evaluation. When the three servers received a feature vector, respective feature values will be compared with each value v_i in non-leaf node N_i. For each edge in decision tree, S_0 and S_1 will obliviously set their cost to 0 if the edge is selected according to the comparison; otherwise, set to a random non-zero value, as depicted in Figure 4. Then, S_0 and S_1 jointly sum up edge costs for all paths. Among all costs of paths, only one is zero, that is, the corresponding path is the decision process and the classification of the leaf node in this path is detection result.

As described in Protocol 2, the process of evaluation contains three key steps: feature selection, comparison, and path evaluation.

4.4.1. Feature Selection. For each node $N_i = \{t_i, v_i\}$, t_i is stored in S_0 and S_1 in the shared form $[t_i]$. $[t_i]$ will be extended to $\langle t_i \rangle = \{(t_i)_0, (t_i)_1, 0\}$, and S_2 holds 0. Then, run 3-party shared OT mentioned above to get the feature value $\langle x_{t_i} \rangle$.

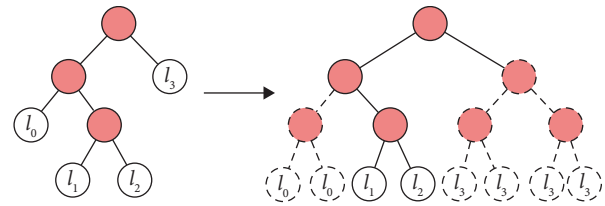


Figure 3: Padding binary tree.

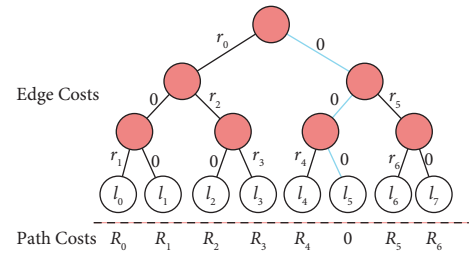


Figure 4: Each edge is set to 0 or random value r_i ($i = 0, 1, 2, \ldots$). Each path to a leaf node has a cost R_j ($j = 0, 1, 2, \ldots$) by summing up all costs of edges in the path, and only one's cost is 0.

4.4.2. Comparison. Comparison depends on the DICF scheme, where S_2 generates keys and S_0, S_1 are evaluators. S_2 generates a pair of keys for each non-leaf node $N_i = \{t_i, v_i\}$ to compare corresponding feature value of input with a random value μ_i, such that servers cannot obtain v_i. Then, S_0, S_1 get a value $\Delta x_i = x_{t_i} - v_i + \mu_i$ by jointly computing. Next, S_0 and S_1 jointly get comparison result $[b_i]$ by evaluating DICF keys with Δx_i, as $b_i = 0$ if $(x_{t_i} - v_i) \leq 0$ and $b_i = 1$ otherwise.

4.4.3. Path Evaluation. S_0 and S_1 generate random value r_i together for each non-leaf node N_i in the tree. Then, S_0 and S_1 locally compute the left out-going edge cost $[e_{i,\text{left}}] = [b_i \cdot r_i]$ and the right-going edge cost $[e_{i,\text{right}}] = [(1 - b_i) \cdot r_i]$ for node N_i. Then, as depicted in Figure 4, S_0 and S_1 jointly get path costs $[\mathbf{P}]$, where $\mathbf{P} := (p_0, p_1, \ldots, p_{\mathcal{L}-1})$ and only one path cost in $\mathbf{P}$ is 0. To obliviously get classification result according to the position of 0 in $\mathbf{P}$, S_0, S_1 jointly pick a random value $\delta \longleftarrow \mathbb{Z}_{\mathcal{L}}$, and cyclic shift $\mathbf{L}$ and $\mathbf{P}$ to the

Initialization:
  for *each* $j \in \mathbb{Z}_3$ **do**
    S_j and S_{j+1} agree on the same random seed $\varphi_j \longleftarrow \{0,1\}^\tau$;
  end
Preparing:
  $\varepsilon = 2^{\ell-1} - 1$;
  for *each* $i \in \mathbb{Z}_{\mathcal{N}}$ **do**
    S_2 generates $\mu_i \longleftarrow \mathbb{Z}_{2^\ell}$;
S_2 generates keys $((\mathcal{F}_{\mu_i})_0, (\mathcal{F}_{\mu_i})_1)$ for $\mathbf{F}_{0,\varepsilon,\mu_i,0}$: $\mathbb{Z}_{2^\ell} \longrightarrow \mathbb{Z}_{2^\tau}$;
    S_2 sends $(\text{sid}, (\mathcal{F}_{\mu_i})_0)$ to S_0, $(\text{sid}, (\mathcal{F}_{\mu_i})_1)$ to S_1;
  end
for *each* S_j **do**
  for *each* $i \in \mathbb{Z}_{\mathcal{N}}$ **do**
    $r_{i,j} \longleftarrow \text{PRF}_{\varphi_j}^{\mathbb{Z}_{2^\ell}}(\text{sid}, i)$, $r_{i,j+2} \longleftarrow \text{PRF}_{\varphi_{j+2}}^{\mathbb{Z}_{2^\ell}}(\text{sid}, i)$;
    $(t_i)_2 \longleftarrow 0$, $(v_i)_2 \longleftarrow \mu_i$;//feature selection
    Run 3-party shared OT protocol to get $(x_{t_i})_j$;
    $(\Delta x_i)_j \longleftarrow (x_{t_i})_j - (v_i)_j + r_{i,j} - r_{i,j+2} \pmod{2^\ell}$;//comparison
  end
  $(\Delta x)_j := ((\Delta x_0)_j, (\Delta x_1)_j, \ldots, (\Delta x_{\mathcal{N}-1})_j)$;
  Send $((\text{sid}, (\Delta \mathbf{x})_j)$ to S_0, S_1;
end
for each $j \in \{0,1\}$ **do**
  S_j receives $(\text{sid}, (\Delta \mathbf{x})_{1-j})$ from S_{1-j}, $(\text{sid}, (\Delta \mathbf{x})_2)$ from S_2;
  for *each* $i \in \mathbb{Z}_{\mathcal{N}}$ **do**
    $\Delta x_i \longleftarrow (\Delta x_i)_0 + (\Delta x_i)_1 + (\Delta x_i)_2 \pmod{2^\ell}$;
    $(b_i)_j \longleftarrow \text{DICF.Eval}_{0,\varepsilon}(j, (\mathcal{F}_{\mu_i})_j, \Delta x_i)$;
    $r_i \longleftarrow \text{PRF}_{\varphi_0}^{\mathbb{Z}_{2^\tau}}(\text{sid}, i)$;//path evaluation
$(e_{i,\text{right}})_j \longleftarrow (1 - j - (b_i)_j) \cdot r_i$, $(e_{i,\text{left}})_j \longleftarrow (b_i)_j \cdot r_i$;
  end
  $\delta \longleftarrow \text{PRF}_{\varphi_0}^{\mathbb{Z}_{\mathcal{L}}}(\text{sid}, 0)$;
  for $i \in \mathbb{Z}_{\mathcal{L}}$ **do**
    $(p_i)_j \longleftarrow$ Sum up the share of edge costs along i-th leaf node's path;
    $(p_i')_j \longleftarrow (p_{i-\delta \pmod{\mathcal{L}}})_j$;
    $(q_i)_0 \longleftarrow \text{PRF}_{\varphi_0}^{\mathbb{Z}_{2^\ell}}(\text{sid}, i, 0)$, $(q_i)_1 \longleftarrow \text{PRF}_{\varphi_0}^{\mathbb{Z}_{2^\ell}}(\text{sid}, i, 1)$;
    $(l_i'')_j \longleftarrow (l_{i-\delta \pmod{\mathcal{L}}})_j - (q_i)_j \pmod{2^\ell}$;
  end
    $(\mathbf{P}')_j := ((p_0')_j, (p_1')_j, \ldots, (p_{\mathcal{L}-1}')_j)$, $((\mathbf{L}'')_j := (l_0'')_j, (l_1'')_j, \ldots, (l_{\mathcal{L}-1}'')_j)$;
    Send $(\text{sid}, (\mathbf{P}')_j, (\mathbf{L}'')_j)$ to S_2
end
S_2 receives $(\text{sid}, (\mathbf{P}')_0, (\mathbf{L}'')_j)$ from S_0, $(\text{sid}, (\mathbf{P}')_1, (\mathbf{L}'')_1)$ from S_1;
for *each* $i \in \mathbb{Z}_{\mathcal{L}}$ **do**
  if $(p_i)_0 + (p_i)_1 = 0 \pmod{2^\tau}$ **then**
    $\omega \longleftarrow i$;
    $((\mathcal{F}_\omega)_0, (\mathcal{F}_\omega)_1 \longleftarrow \text{DPF.Gen}(1^\tau, f_{\omega,1})$ for point function $f_{\omega,1}$: $\mathbb{Z}_{\mathcal{L}} \longrightarrow \mathbb{Z}_{2^\ell}$;
    Send $(\text{sid}, (\mathcal{F}_\omega)_0)$ to S_0, $(\text{sid}, (\mathcal{F}_\omega)_1)$ to S_1;
    Return $l_\omega'' := (l_\omega'')_1 + (l_\omega'')_2 \pmod{2^\ell}$;
  end
end
for *each* $j \in \{0,1\}$ **do**
  S_j receives $(\text{sid}, (\mathcal{F}_\omega)_j)$;
$\{(\beta_0)_j, (\beta_1)_j, \ldots, (\beta_{\mathcal{L}-1})_j\} \longleftarrow \text{DPF.EvalAll}(j, (\mathcal{F}_\omega)_j)$;
  $(q_\omega)_j \longleftarrow \sum_{i=0}^{\mathcal{L}-1} (((q_i)_0 + (q_i)_1) \cdot (\beta_i)_j) \pmod{2^\ell}$;
  Return $(q_\omega)_j$;
end

Protocol 2: Constant-round evaluation protocol.

right δ position to obtain $\mathbf{L}' = (l_0', l_1', \ldots, l_{\mathcal{L}-1}')$ and $\mathbf{P}' = (p_0', p_1', \ldots, p_{\mathcal{L}-1}')$. Then, they generate a random vector $\mathbf{Q}$: $= (q_0, q_1, \ldots, q_{\mathcal{L}-1}) \longleftarrow (\mathbb{Z}_{2^\ell})^{\mathcal{L}}$, and jointly compute $\mathbf{L}'' = \mathbf{L}' - \mathbf{Q}$ (l_i'': $= l_i' - q_i$ is the element with index i in $\mathbf{L}''$). Subsequently, S_0, S_1 reveal $\mathbf{P}', \mathbf{L}''$ to S_2. After obtaining $\mathbf{P}', \mathbf{L}''$, S_2 generates a pair of DPF keys for point function $f_{\omega,1}(x)$, where ω is the index of $p_\omega' = 0$. Lastly, S_0, S_1 serve as evaluators and jointly compute $[q_\omega] = \sum_{i=0}^{\mathcal{L}-1} (q_i \cdot$

$[f_{\omega,1}(i)])\ (\mathrm{mod}\ 2^{\ell})$. S_0, S_1 send $[q_\omega]$ to ICS and S_2 sends l''_ω to ICS, so that the system admin of ICS gets classification result. Note that the result $l'_\omega = q_\omega + l''_\omega\ (\mathrm{mod}\ 2^{\ell})$.

4.5. Security Analysis. The main building block of our privacy-preserving decision tree evaluation protocol is the 3-party shared OT (cf. Section 4.1). The construction of the 3-party shared OT protocol is inspired by Paul et al. [14]. At high level, in turn, each of the three servers plays the role of a DPF generator, and the other two servers play the role of DPF evaluators to obtain the i-th position value of their (2, 3)-replicated shared data $\mathbf{x}$. In particular, for instance, when S_0 is the DPF generator, it generates a pair of DPF keys for a random position $\mu \in \mathbb{Z}_n$. Let the shared OT choice be $[i]$. In the online phase, the servers jointly open $\delta := i - \mu\ (\mathrm{mod}\ n)$ to S_1, S_2. They can then shift the shared data $\mathbf{x}$ by δ position such that the μ-th position of the shifted data $\mathbf{x}'$ is x_i.

We now analyze the security of this part. First of all, revealing $\delta := i - \mu\ (\mathrm{mod}\ n)$ leaks no information about i, as i is masked by μ information theoretically. Moreover, assuming that the underlying DPF scheme is secure, S_1 and S_2 obtain the shared form of x_i. Repeating the above process for all three servers, we obtain the final result. Note that, for efficiency, we use PRF to generate shares of 0 without communication; assume that the underlying PRF is secure, and the generated shares are computationally indistinguishable from uniformly random ones. Therefore, when any of the three servers is semi-honest corrupted, its view is computationally indistinguishable from a few random shares (and the DPF keys).

When the model is not a full binary tree, we pad the model to a full binary tree by adding dummy nodes; therefore, the tree evaluation process does not leak any information about the tree structure to the MPC players. The security of the feature selection phase can be reduced to the security of the 3-party shared OT. In addition, we adopt the DICF scheme for secure comparison, and its security is proven in [11]. Finally, with regard to the path evaluation, we designed an encoding scheme for the tree such that we can evaluate the path within one multiplicative round. As a result, after evaluation, only the output label will be 0, and the remaining labels are uniformly random. Since each tree uses different fresh random encoding instances, the three severs cannot learn any additional information other than the intended output label.

5. Implementation and Benchmark

5.1. Dataset Description and Experiment Setup. In the experiment, we adopt CICIDS2017 [15] as dataset. It captures normal packages and attacks in simulated network environment that is similar to the real-world network. The dataset contains several CSV files, and each of them includes a kind of attack. We perform experiment on the CSV files corresponding to DDoS, DoS, botnet, port scan, and web attacks, as described in Table 2. Each entry in CSV file contains a feature vector and a class label. The feature vector includes 78 features, such as timestamp, source IP, destination IP, package length, and protocol.

Table 2: Description of CSV files of CICIDS2017.

Attack type	Normal samples	Attack samples	Total samples
DoS	440031	252661	692692
DDoS	97718	128027	225745
Botnet	189067	1966	191033
Port scan	127537	158930	286467
Web attacks	168186	2180	170366
Total	1022539	543764	1566303

We evaluate the performance of the GBDT model and CART model for the binary classification task with CICIDS2017 dataset. The samples of DDos, DoS, botnet, port scan, and web attacks are labeled as attack, and the others are labeled as normal. We randomly select 80% of this dataset as our training set and the remainder as validation set.

5.2. Evaluation Metrics. To evaluate the performance of intrusion detection, we adopt some evaluation metrics, such as Precision, Recall, and F_1. These metrics depend on four parameters. (i) True Positive (TP) denotes the number of attack samples that are correctly classified. (ii) False Negative (FN) denotes the number of attack samples that are wrongly classified. (iii) True Negative (TN) denotes the number of normal samples that are correctly classified. (iv) False Positive (FP) denotes the number of normal samples that are wrongly classified.

(i) Precision indicates how many samples that are classified as attacks are real attacks.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{8}$$

(ii) Recall indicates how many attack samples are correctly classified. Since the proportion of attack in the total sample is small and the attack will cause severe consequence, we need to identify as many attacks as possible. Therefore, Recall is an important evaluation metric.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{9}$$

(iii) F_1-score is calculated based on Precision and Recall and shows the trade-off between Precision and Recall.

$$F_1 = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{10}$$

5.3. Intrusion Detection Result. In our experiment, we utilize GBDT mentioned in Section 2.2 as our detection model firstly. We set the regularization coefficients $\gamma = 1$ and $\lambda = 1$ and learning rate as 0.1. In the GBDT model, each tree's maximum depth is set to 9. Figure 5 shows the performance of GBDT model, when the iterations are 5, 10, 15, 20, 40, and 60.

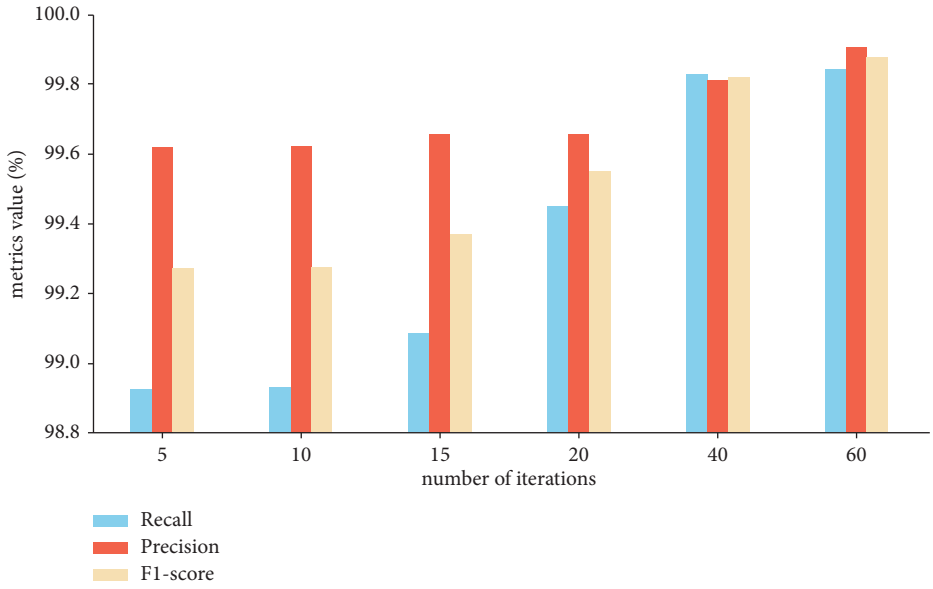


Figure 5: The anomaly detection result using GBDT model with different number of iterations to detect samples in validation set.

We use regularization item to prevent overfitting. As shown in Figure 5, the metrics Recall and F_1-score notably enhance as the number of iterations increases. When the number of iterations reaches 60, the GBDT model performs good on three metrics (99.84% Recall, 99.90% Precision, and 99.87% F_1).

Then, we use a CART decision tree model as detection model to evaluate its performance on CICIDS2017 dataset. We set the maximum depth of decision tree to 9 and obtain its evaluation result (99.09% Recall, 94.32% Precision, and 96.65% F_1). By adopting multiple decision trees for iterative learning, the GBDT model obtain stronger classification ability than single CART decision tree.

5.4. Time Efficiency. We run our platform in different network environments to evaluate its time efficiency. The network environments are simulated, including LAN (0.1 ms RTT, 1 Gbps bandwidth), MAN (6 ms RTT, 100 Mbps bandwidth), and WAN (80 ms RTT, 40 Mbps bandwidth). We set the depth of full decision tree to 5, 7, 9, 11, and 13. We evaluate the time efficiency of our proposed platform when the platform evaluates one tree and evaluates one thousand trees. Each tree is a full binary tree. Our benchmarks are executed on a desktop with Intel(R) Core i7 8700 CPU @ 3.2 GHz, and the operating system is Ubuntu 18.04.2 LTS with 6 CPUs, 32 GB memory, and 1 TB SSD.

As shown in Table 3, our platform performs good in different simulated network environments. Our platform can evaluate one thousand trees whose depth is 9 in 11 seconds when the network environment is LAN (0.1 ms RTT, 1 Gbps bandwidth). However, the increasing depth of decision tree results in more communication cost because the constant-round protocol's communication cost is $O(2^d)$, where d is depth of full tree. Therefore, the proposed protocol is not suitable for the tree model whose depth is greater than 9. As GBDT uses multiple trees, the GBDT model is significantly slower than the CART model. Each tree of GBDT can be evaluated independently, and finally client sums up trees' evaluation result to obtain classification result. Therefore, we can improve the parallel computing capability of the proposed platform to enhance time efficiency of the GBDT model.

6. Related Work

Anomaly detection has been developed for decades and is widely used as defensive method in conventional network. However, since ICS is different from conventional network system, anomaly detection technique cannot be used in ICS directly. Availability and real-time performance are required in ICS-specific IDS [16]. There are a large number of works on ICS-specific IDS. With the development of machine learning (ML) and deep learning (DL) algorithms, most recent works use them to detect anomaly in ICS. The authors in [17] evaluated several machine learning models on an ICS dataset called Power System Dataset, such as Nearest Neighbor, Random Forests, Naive Bayes, SVM, AdaBoost, and JRip. In [18], the authors evaluated different ML and DL algorithms using their generated ICS dataset Electra. These algorithms contain One-Class SVM, SVM, Isolation Forest, Random Forest, and Neural Network. In [19], the authors used the Pearson Correlation Coefficient (PCC) to select packet features and used the Gaussian Mixture Model (GMM) to transform important features for privacy preservation. Then, they used the transformed features as input of a Kalman Filter to detect anomaly. In [20], they utilized Bloom filter to store the signature database for packet-based intrusion detection and applied an LSTM model to learn temporal features.

In private branching program (BP) and decision tree evaluation with constant communication round, there have been several works. The work in [21] evaluates BP with input encrypted by homomorphic public-key cryptosystem.

Table 3: Time efficiency (ms) of the proposed platform when setting different tree depths in different network environments.

Maximum depth		5	7	9	11	13
Evaluating 1 tree	1000mbps/0.1ms	5.1	6.3	9.3	28.1	28.8
	100mbps/6ms	50.6	52.2	55.6	71.4	72.2
	40mbps/80ms	408.9	409.2	410.0	425.7	476.1
Evaluating 1000 trees	1000mbps/0.1ms	4226.9	6713.2	11002.4	29790.6	31540.4
	100mbps/6ms	51990.1	52644.1	54541.3	73736.8	75969.1
	40mbps/80ms	385858.2	391692.5	406701.5	413470.7	413940.0

The maximum depth of tree is set to 5, 7, 9, 11, and 13.

However, it is impractical when the input feature vector is too large. After that, some evaluation protocols with constant communication round are proposed. In [22], the authors utilized additive homomorphic encryption (AHE) and OT for obliviously feature selection and converted the BP model into a secure program with Garble Circuits for comparison. Bost et al. [23] evaluated a decision tree with costly fully homomorphic encryption (FHE) by treating decision tree as a high-degree polynomial. The authors of [24] used OT to select leaf node and DGK protocol based on AHE instead of FHE for comparison. Raymond et al. [25] improved the work in [24] by representing decision tree as linear functions instead of high-degree polynomial form. They computed "path cost" of each leaf node and used it to decide which leaf node contains classification result. In [26], they reviewed prior constant-round approaches and proposed a modular construction from three constant-round sub-protocols: private feature selection, secure comparison, and oblivious path evaluation.

7. Conclusion and Future Work

In this paper, we proposed a privacy-preserving anomaly detection platform for industrial control system. It depends on two main components, detection model and MPC protocol. We use GBDT and CART as anomaly detection models, which are able to detect anomaly with high accuracy. As information privacy is protected by laws and regulations in many countries, we adopt a MPC protocol that can detect network packages from ICS based on decision tree when sensitive data are invisible. The experimental results indicate that the proposed platform can detect anomaly on package level in real time with high accuracy.

Our platform can be developed in several ways in the future. Firstly, we plan to evaluate the performance of our platform in a simulated environment that resembles real environment. In addition, to make detection model more practical, it is necessary to use real data of ICS as training set. Lastly, we will utilize a privacy-preserving machine learning approach in training stage to ensure training data privacy.

Data Availability

The experiment data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Key R&D Program of China (no. 2021YFB3101601), the National Natural Science Foundation of China (grant no. 62072401), and the Open Project Program of Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province. This project was also supported by Input Output (iohk.io).

References

[1] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *Proceedings of the IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4490–4494, IEEE, Melbourne, Australia, November, 2011.

[2] A. Hobbs, *The Colonial Pipeline Hack: Exposing Vulnerabilities in Us Cybersecurity*, SAGE Publications, Thousand Oaks, CA, USA, 2021.

[3] P. Voigt and A. Bussche, "Practical implementation of the requirements under the gdpr," in *The EU General Data Protection Regulation (GDPR)*, pp. 245–249, Springer, Berlin, Germany, 2017.

[4] E. Goldman, *An Introduction to the california Consumer Privacy Act (Ccpa)*, Santa Clara Univ. Legal Studies Research Paper, Santa Clara, CA, USA, 2020.

[5] K. Xu, "The effectiveness and function of the personal information security specification," *China Information Security*, vol. 4, no. 3, 2019.

[6] H. Gao, L. Yuan, F. Yin, and G. Shen, "Epcad: efficient and privacy-preserving data anomaly detection scheme for industrial control system networks," in *Journal of Physics: Conference Series*vol. 1856, IOP Publishing, Article ID 012028, 2021.

[7] A. C.-C. Yao, "How to generate and exchange secrets," in *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pp. 162–167, IEEE, Toronto, Canada, October, 1986.

[8] Y. Huang, D. Evans, and J. Katz, *Private Set Intersection: Are Garbled Circuits Better than Custom Protocols?* NDSS, Manhattan, NY, USA, 2012.

[9] P. Mohassel and Y. Zhang, "Secureml: a system for scalable privacy-preserving machine learning," in *Proceedings of the 2017 IEEE symposium on security and privacy (SP)*, pp. 19–38, IEEE, San Jose, CA, USA, May, 2017.

[10] E. Boyle, N. Gilboa, and Y. Ishai, "Function secret sharing," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 337–367, Springer, Berlin, Heidelberg, April, 2015.

[11] E. Boyle, N. Chandran, N. Gilboa et al., "Function secret sharing for mixed-mode and fixed-point secure computation," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 871–900, Springer, Cham, Switzerland, June, 2021.

[12] M. O. Rabin, *How to Exchange Secrets with Oblivious Transfer*, Cryptology ePrint Archive, New York, NY, USA, 2005.

[13] R. G. Bace and P. Mell, "Intrusion detection systems," 2001, http://csrc.nist.gov/publications/nistpubs/800-31/sp800-3% 201.pdf.

[14] B. Paul, J. Katz, E. Kushilevitz, and R. Ostrovsky, "Efficient 3-party distributed oram," in *Proceedings of the International Conference on Security and Cryptography for Networks*, pp. 215–232, Springer, Cham, Switzerland, September, 2020.

[15] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108–116, 2018.

[16] B. Zhu and S. Sastry, "Scada-specific intrusion detection/prevention systems: a survey and taxonomy," *Proceedings of the 1st workshop on secure control systems (SCS)*, vol. 11, no. 7, 2010.

[17] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 3104–3113, 2015.

[18] A. L. Perales Gomez, L. Fernandez Maimo, A. Huertas Celdran et al., "On the generation of anomaly detection datasets in industrial control systems," *IEEE Access*, vol. 7, pp. 177460–177473, 2019.

[19] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 1, pp. 66–79, 2021.

[20] F. Cheng, T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and lstm networks," in *Proceedings of the 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 261–272, IEEE, Denver, CO, USA, June, 2017.

[21] Y. Ishai and A. Paskin, "Evaluating branching programs on encrypted data," in *Proceedings of the Theory of Cryptography Conference*, pp. 575–594, Springer, Berlin, Heidelberg, August, 2007.

[22] J. Brickell, D. E. Porter, V. Shmatikov, and E. Witchel, "Privacy-preserving remote diagnostics," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 498–507, Alexandria, VI, USA, October, 2007.

[23] R. Bost, R. Ada Popa, S. Tu, and S. Goldwasser, *Machine Learning Classification over Encrypted Data*, Cryptology ePrint Archive, New York, NY, USA, 2014.

[24] D. J. Wu, T. Feng, M. Naehrig, and K. E. Lauter, "Privately evaluating decision trees and random forests," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 335–355, 2016.

[25] K. H. Raymond, J. P. K. Ma, Y. Zhao, and S. M. Sherman, "Privacy-preserving decision trees evaluation via linear functions," in *European Symposium on Research in Computer Security*, pp. 494–512, Springer, Berlin, Germany, 2017.

[26] A. Kiss, M. Naderpour, J. Liu, N. Asokan, and T. Schneider, "Sok: modular and efficient private decision tree evaluation," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 2, pp. 187–208, 2019.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 3624587, 14 pages
https://doi.org/10.1155/2022/3624587

Research Article

Weak PassPoint Passwords Detected by the Perimeter of Delaunay Triangles

Lisset Suárez-Plasencia,[1] **Carlos Miguel Legón-Pérez**,[1] **Joaquín Alberto Herrera-Macías**,[1] **Raisa Socorro-Llanes**,[2] **Omar Rojas**,[3,4] **and Guillermo Sosa-Gómez**[3]

[1]*Universidad de La Habana, Facultad de Matemática y Computación, Instituto de Criptografía, Habana 10400, Cuba*
[2]*Universidad Tecnológica de La Habana, Facultad de Informática, Habana, Cuba*
[3]*Universidad Panamericana, Facultad de Ciencias Económicas y Empresariales, Álvaro Del Portillo 49, Zapopan, Jalisco 45010, Mexico*
[4]*Faculty of Economics and Business, Universitas Airlangga, Surabaya, East Java 60286, Indonesia*

Correspondence should be addressed to Guillermo Sosa-Gómez; gsosag@up.edu.mx

Received 31 January 2022; Revised 27 June 2022; Accepted 11 July 2022; Published 21 September 2022

Academic Editor: Sridhar Adepu

PassPoint is a graphical authentication technique that is based on the selection of five points in an image. A detected vulnerability lies in the possible existence of a pattern in the points that make up the password. The objective of this work is to detect nonrandom graphical passwords in the PassPoint scenario. A spatial randomness test based on the average of Delaunay triangles' perimeter is proposed, given the ineffectiveness of the classic tests in this scenario, which only consists of five points. A state-of-the-art of various applications of Voronoi polygons and Delaunay triangulations are presented to detect clustered and regular patterns. The distributions of the averages of the triangles' perimeters in the PassPoint scenario for various sizes of images are disclosed, which were unknown. The test's decision criterion was constructed from one of the best distributions to which the data were adjusted. Type I and type II errors were estimated, and it was concluded that the proposed test could detect clustered and regular graphical passwords in PassPoint, therefore being more effective in detecting clustering than regularity.

1. Introduction

Graphical authentication schemes are alternatives to passwords based on alphanumeric characters. These are used in user authentication or key generation for use in cryptographic algorithms [1]. Graphic passwords can be formed by the combination of photos, images, or iconography. Given the characteristics of the images, they produce a much larger password space and are more resistant to dictionary attacks since alphanumeric password phrases that are relatively easy to predict are often used. These passwords' efficiency is based on the ability of humans to remember patterns in images instead of memorizing sets of characters of great length and complexity.

An updated description and critical assessment of the different graphical authentication schemes' security and usability can be found in [2]. PassPoint is a graphical authentication technique that bases its operation on selecting and remembering patterns of points in images [3]. The authentication process involves the user selecting various points on the image in a particular order. When logging in, the user is supposed to click near the points selected in the registration phase within a tolerance region or neighborhood. One of the vulnerabilities of PassPoint lies in the possible existence of a pattern in the points that make up the password [2]. This pattern can be determined either by selecting the points or by the spatial distribution of them in the image. Considering the latter, a password is considered weak if the points are not randomly distributed and can be obtained by an attacker applying various techniques such as those described in [4–7]. The main types of nonrandomness present between the points, in that case, are clustering,

regularity, and smoothness. According to the behavior of the points distributed in the plane (in this case, image), the spatial point patterns are classified into random (homogeneous Poisson point process), regular (uniform or a pattern in inhibition), or clustered (aggregates), [8–12]. During the registration phase of the PassPoint, it is necessary to determine whether the points selected by the user follow a random spatial pattern.

In [13], it is stated that Delaunay triangulation and Voronoi polygons have been widely used to analyze the pattern of distribution of points and measure spatial intensity. To measure the distribution of points, we calculate the nearest neighbor and the point pattern shape. When calculating a Voronoi diagram to a point distribution to test the complete spatial randomness of the point distributions, the characteristics of the Delaunay triangles are extracted (e.g., interior angles and edge lengths). Spatial intensity, i.e., how concentrated the points are in a particular study area, is measured by calculating the area and elongation of the Voronoi polygons. This approach has been used in many applications, including agriculture, microbiology, and astronomy [14].

In this work, a statistical test is proposed to detect clustering or regularity between the points of a graphical password in PassPoint. This test is based on the Delaunay triangles generated by that password, specifically on the average of those triangles' perimeters. The effectiveness of the proposed test is experimentally verified. Type I error resulting when applying them to random passwords is estimated and kept at acceptable levels for practical applications; on the other hand, type II error resulting when applied to clustered and regular passwords is estimated, and as expected, it is observed that it depends on the level of clustering or regularity. The article is structured in 4 sections: Section 1 shows the Introduction; Section 2 is composed of PassPoint, spatial point patterns, classic tests most used in complete spatial randomness, and the applications of Voronoi diagrams and Delaunay triangulations in the detection of spatial point patterns. Section 3 shows our contribution: detection of weak graphical passwords in PassPoint, based on the perimeter of their Delaunay triangles, and finally in Section 4, the conclusions and future work are presented.

2. Preliminaries

2.1. PassPoint. PassPoint is a graphical authentication scheme of the cued-recall type presented in [3]. This technique requires the user to select as their password during the registration phase an ordered set of 5 points (pixels) in an image. In the authentication phase, the same points must be selected approximately and in the same order that they were registered. For the authentication process to be effective and convenient for the user, there must be a tolerance associated with each point (approximately 0.25 cm). It is possible to use any image to select the password points; it can be provided by the user or the system itself. The authors of this scheme recommend using images that have hundreds of Hotspots spread evenly for greater security. The password is not stored explicitly, but a hash of the concatenation of the password points is generated. However, this causes a problem when applying the password hashing function. It is unlikely that the user will select the same points selected in the authentication phase-image in the registration phase, which means that the password hashing function will always be different. To establish the tolerance around each point, a discretization mechanism is used, which reduces the password space and provides relevant information to carry out a dictionary attack [15]. A discussion about the importance of the discretization mechanism in graphic password schemes can be seen in [16–18], while in [16–19], some of the different methods of discretization known so far are presented.

While the selection of images by the user may increase the ability to memorize their password, there is a possibility that, at the same time, security will be compromised with images with few security features (e.g., few memorable points and images that are easy to predict with knowledge about the user) [3]. In several studies such as those presented in [7, 15, 20, 21], dictionary attacks have been carried out using digital image processing techniques. The spatial patterns in the user's selection of points reduce the effective space of a password and give an advantage to possible attackers, who can use this knowledge to increase their attacks' probability of success. In the study presented by [22], it is suggested that it is possible to obtain patterns in the shape and order of the selection of the points without knowing the image used to create the password. Users tend to select their password points in separate compositions from the background images, to facilitate the memorability of their passwords. If the set of points selected by the user as their graphical password does not follow a random pattern, it presents a shape of a straight line, curved or by default (Z, W, C, V), or of every 2 consecutive points out of the 5 that make up the password; they are at constant distances. Then, said graphical password is considered weak, as it can be compromised using dictionary attacks [2, 5, 23].

2.2. Spatial Point Patterns. The phenomena that occur in some regions of space, such as data on human settlements, animals, the cultivation of crops, or information on the behavior of a pandemic (such as COVID-19 in 2020), represent an occurrence through its spatial coordinates (x, y). The datasets generated by these coordinates are called spatial point patterns [8, 10, 11, 24, 25]. From the study of spatial patterns, inferences can be made about the existence of interactions between each population's individuals. Spatial point patterns are classified as random (homogeneous Poisson point process), regular (uniform or an inhibiting pattern), or clustered (aggregated); see Figure 1.

To decide the behavior of an observed point pattern, a complete spatial randomness (CSR) test is applied where it is assumed as a null hypothesis that the pattern comes from the Poisson distribution; that is, that the pattern of points follows a random distribution [8, 26, 27]. The spatial point patterns present two fundamental characteristics [12, 27]. One of them is related to the intensity of the number of points per unit area; the second is based on looking for

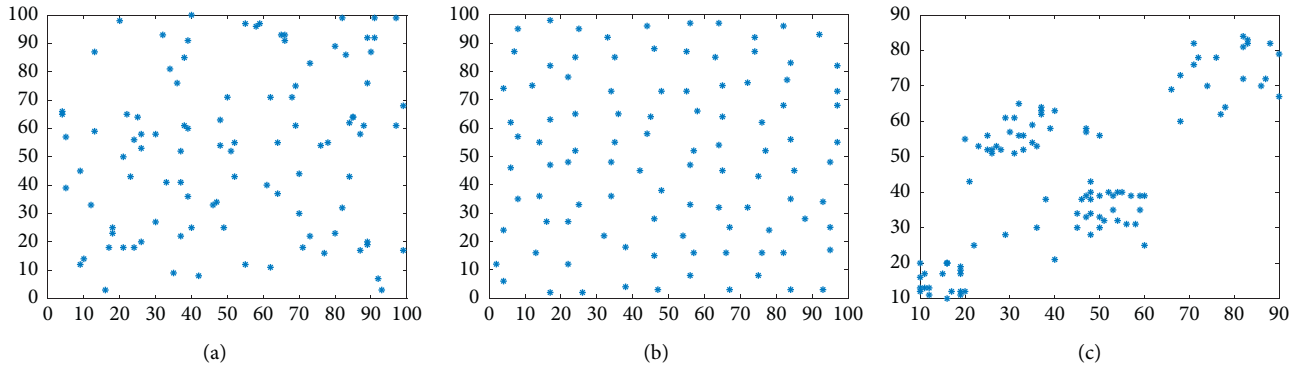


Figure 1: Random point pattern: (a) regular (b) and clustered (c).

relationships between each point with those of its surroundings, mainly through the distance between points.

2.3. Classic Tests Most Used in Complete Spatial Randomness (CSR)

2.3.1. K-Ripley Function. If a Poisson process randomly distributes a set of points with intensity λ, the expected number of points in a circle of radius r is $\lambda\pi r^2$. The deviation from randomness can be quantified using the K-Ripley function [8, 25, 27], which reflects the type, intensity, and range of the spatial pattern by analyzing the distances between the points, defined as follows:

$$K(r) = \frac{A}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k_{i,j}(r)e_{i,j}(r), \tag{1}$$

for all $i \neq j$, where n is the number of points in the pattern, A is the area of the region under study, $e_{i,j}(r)$ is the edge correction method, and $k_{i,j}(r)$ is the following indicator function:

$$k_{i,j}(r) = \begin{cases} 1, & \text{if } r_{i,j} \leq r, \\ 0, & \text{if } r_{i,j} > r, \end{cases} \tag{2}$$

where $r_{i,j}$ is the distance between points i and j. The edge effects arise because the points that appear outside the limits of the study area are not taken into account to estimate the statistic, even though they are at a distance less than r from a point located within the area. One of the possible expressions of the K-Ripley function, taking into account one of the edge correction methods, is as follows:

$$K_{\text{bord}}(r) = \frac{A\sum_{i=1}^{n}\xi_i(r)\sum_{j=1}^{n}k_{i,j}(r)}{n\sum_{i=1}^{n}\xi_i(r)}, \tag{3}$$

where ξ_i denotes the indicator function that is equal to 1 if the distance from a point p_i to the edge A is greater than or equal to r and 0 otherwise. It is worth clarifying that there are other ways to correct the edge effect, which lead to alternative expressions of the K function. A detailed review of these methods can be found in [8, 28].

The transformation $\widehat{L}(r) = \sqrt{K(r)/\pi}$ allows linearizing the function $K(r)$ and stabilizing the variance, and by means of the $L(r) = \widehat{L}(r) - r$ transformation, it is possible to adjust the Poisson pattern to the value of zero. A clustered pattern occurs when $L(r)$ is significantly greater than zero, and a regular pattern occurs when $L(r)$ is significantly less than zero.

2.3.2. The G Function, Distance to the Nearest Neighbor. This method is based on the distances from each point to its nearest neighbor [8, 27]. The expected cumulative distribution function for the nearest neighbor distances d is defined by the Poisson distribution:

$$G(d) = 1 - e^{-\lambda\pi d^2}. \tag{4}$$

If over an area A, n points are randomly distributed, where $\lambda = n/A$. To consider the correction of the edge effect, the following function is used:

$$\widehat{G}(d) = \frac{\sum_{i=1}^{n} I_i(d)}{n}, \tag{5}$$

where n is the number of points in the pattern and $I_i(d)$ is the indicator function, which takes the value of one if the Euclidean distance between point i and its closest neighbor is less than d, and 0 otherwise; see [8]. A clustered pattern occurs when $\widehat{G}(d) > G(d)$, while a regular pattern occurs when $\widehat{G}(d) > G(d)$.

2.3.3. The Function F, Distance to the Null Space. The null space distance measures the distance d from each point in an additional m set, called a grid, to the closest of the n points in the observed pattern. For a pattern under the CSR hypothesis, its distribution is the same as for the function $G(d)$, i.e.,

$$F(d) = G(d) = 1 - e^{-\lambda\pi d^2}, \tag{6}$$

where λ is the intensity of the pattern. For estimating distances, a set of m points similar to n of the observed pattern is usually used. The distribution of the observed pattern is estimated by

$$\widehat{F}(d) = \frac{\sum_{j=1}^{m} I_j(\mathrm{d})}{m}, \tag{7}$$

where m is the number of points on the grid and $I_j(d)$ is the indicator function that the value of one if the Euclidean distance between point j on the grid and its closest neighbor in the pattern is less than d, and 0 otherwise.

The use of the $F(d)$ function is similar to that of the $G(d)$ function, using Monte Carlo simulations to estimate its critical values and graphical diagnostic tools in the same way. However, the interpretation of the deviations from the observed distribution is opposite: values more significant than those of the theoretical distribution indicate regularity and smaller values indicate clustering. The F function is usually more effective at detecting CSR deviations towards the cluster; see [27].

2.4. Applications of Voronoi Diagrams and Delaunay Triangulation in the Detection of Spatial Point Patterns. Voronoi diagrams are geometric structures that allow you to build a partition of the Euclidean plane. Given an initial set $P = \{p_1, p_2, \ldots, p_n\}$ of n points in the plane, a Voronoi diagram is defined as a partition of the Euclidean plane into n disjoint regions.

Definition (a planar ordinary Voronoi diagram): Let $P = \{p_1, p_2, \ldots, p_n\} \subset \mathfrak{R}^2$, where $2 \leq n < \infty$ and $p_i \neq p_j$, for, $i, j \in J_n$. We call the region given by

$$V(p_i) = \{q: q - p_i \|_2 \leq \|q - p_i \|_2, \text{for} j \neq i, j \in J_n\}. \quad (8)$$

The planar ordinary Voronoi polygon associated with p_i (or the Voronoi polygon of p_i), and the set given by

$$V = \{V(p_1), \ldots, V(p_n)\}. \quad (9)$$

The planar ordinary Voronoi diagram by P (or the Voronoi diagram of P): we call p_i of $V(p_i)$ the i th Voronoi polygon, and the set $P = \{p_1, p_2, \ldots, p_n\}$ is the generator set of the Voronoi diagram V (in the literature, a generator point is sometimes referred to as a site). [29].

For the dual graph of a Voronoi diagram is a Delaunay triangulation, see Figure 2. A triangulation of the set P of points on the plane is Delaunay if and only if the circumscribed circumference of any triangle in the lattice does not contain a point of P in its interior. This condition is known as Delaunay's condition. The Voronoi diagrams and the Delaunay triangulation in the two-dimensional case present a series of characteristics determined by the behavior of the point pattern observed in the initial set of points [9, 29, 30].

Since the mid-1980s, some of these characteristics have been used in the study of spatial point patterns. For example, in [31], although the total number of patterns examined is not large, the influence of a Delaunay triangle's interior angles is studied to detect clustering at the points. In general, the authors concluded that the minimum angle seems preferable to the maximum one to detect clustered or regular patterns. However, there are indications that the maximum angle seems to detect some cases of clustering that are not discernible by the minimum angle. In order to analyze whether the

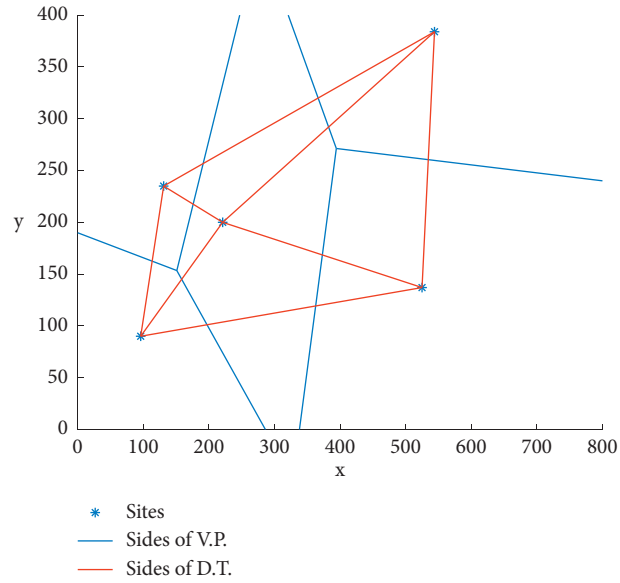


Figure 2: Representation of a Voronoi diagram (VD) and its corresponding Delaunay triangulation (DT).

characteristics, interior angle of a Delaunay triangle, minimum angle, mean angle, and maximum angle of a Delaunay triangle, length of one side of a Voronoi polygon, the distance between a site and a vertex of its Voronoi polygon (radius of a circle circumscribed in a Delaunay triangle), length of one side of a Delaunay triangle, and area and perimeter of a Delaunay triangle are capable of detecting nonrandomness. In [9], they generated 100 clustered or regular points in a square unit. Obtaining the characteristic "minimum angle of a Delaunay triangle" is more effective in detecting regular patterns than the others in detecting clustered patterns. An adaptive spatial clustering algorithm based on Delaunay triangulation is proposed in [32]. This algorithm uses both the Delaunay triangulation edge's statistical characteristics and a new definition of spatial proximity based on the Delaunay triangulation to detect spatial clusters.

Discovery of Spatial Patterns with Extended Objects (DEOSP) [33, 34] is another method that allows for the discovery of patterns for extended objects (straight lines, strings of lines, and collections of the same), although it does not allow operating on the extended objects as areas. DEOSP is based on structures related to the Delaunay triangulation. The work presented in [35] uses the area and perimeter of the Voronoi polygons to analyze changes in the spatial patterns of permanent GNSS (Global Navigation Satellite System) stations ASG-EUPOS (Active Geodetic Network-European Position Determination System) in Poland depending on the scales used. Another vital application of Voronoi polygons is the one presented in [36]. In it, the analysis of macromolecular complexes is presented from a method based on 3D Voronoi tessellations. The method enables local density estimation, segmentation, and quantification of 3D particle localization microscopy data;

specifically, the authors use the area of Voronoi polygons to detect the clustering of particles.

3. Detection of Weak Graphical Passwords in PassPoint, Based on the Perimeter of Their Delaunay Triangles

3.1. Ineffectiveness of the Classic CSR Tests in the PassPoint Scenario. As far as we are aware of, there is no consensus in the current literature on the minimum value of the number of points (n) of the pattern from which the classic tests described in subsection 2.3 are considered effective. In [37], the authors applied the tests to a pattern of 22 points, the smallest pattern of the reference; however, the results achieved are not discussed. Also, in [37], the authors experimented with a pattern of 36 points, for which they concluded that the tests were effective. So we propose the following research question: what will happen in the PassPoint scenario and where are the patterns with only 5 points available?

From the results carried out in [38], it is known that the K-Ripley function tests and those of the distance to the nearest neighbor are ineffective in detecting graphic passwords formed by patterns clustered in PassPoint; however, the experiments were performed for a relatively large number of Monte Carlo simulations. This article analyzes three of the classic tests most used in CSR, including the two tests mentioned above, in detecting nonrandomness in PassPoint passwords, but with a smaller number of Monte Carlo simulations. This difference is given by the existing controversy between the number of simulations in the consulted bibliography, since in [37], the authors state that for a significance level of $\alpha = 0.05$, it is advisable to perform at least 999 simulations, while in [8], they state that for $\alpha = 0.05$ and $\alpha = 0.01$, 40 and 199 Monte Carlo simulations must be performed, respectively.

To analyze the detection of nonrandomness of these tests in the PassPoint scenario, two experiments were carried out on a 1920×1080 pixel image, one to measure clustering and the other regularity. The experiments carried out were run in MATLAB version R2018a on an AMD A6-9220e CPU: 1.60 GHz with 4 G of RAM.

The experiments were designed as follows: for experiment 1, two databases were generated, DB. $1.1_{Ag.(IV)}$ and DB. $1.2_{Ag.(VIII)}$, of 10,000 passwords with Poisson aggregate patterns with an aggregation distance of $686u$ and $315u$, respectively, [37]. That is, two databases of passwords were generated, clustered in an area equivalent to a quarter of the image and the other to an eighth, containing the DB. $1.2_{Ag.(VIII)}$ with a higher level of clustering. The clustered (or aggregated) patterns were derived from a Poisson aggregate process: randomly distributed parental points were generated, and subsequently derived points were randomly distributed around the parents within a specified aggregation radius [8, 37]. For experiment 2, the pattern xy with the highest possible regularity level was generated, which is determined by the following points: $(0; 0)$, $(1920; 0)$, $(0; 1080)$, $(1920; 1080)$, and $(960; 540)$; see Figure 3.

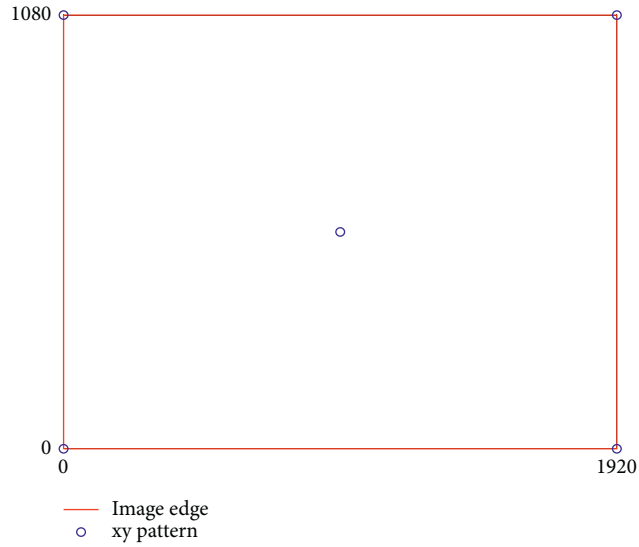


Figure 3: Representation of the xy pattern.

Now we discuss the results obtained after running both experiments. For each of the tests, the critical values were estimated using 199 Monte Carlo simulations of sets of 5 random points on a rectangle of size 1920×1080. In addition to the K-Ripley function, the confidence intervals were estimated according to Ripley's approximation + [27, 39], where $A = 1920 \cdot 1080$ and $n = 5$. These Monte Carlo simulations guarantee critical intervals with a significance level of $\alpha = 0.01$ for each test. See Figure 4, where the continuous line represents the theoretical value of the null hypothesis, the dashed lines represent the critical values of each of the tests in 199 simulations of random patterns. In the case of the K-Ripley function, the dashed lines represent the confidence intervals for $\alpha = 0.01$ of the test according to Ripley's approximation. It is observed how the critical values coincide with the minimum value of each function.

From the estimated critical values, an immediate conclusion was obtained: the K-Ripley function tests and the nearest neighbor are not effective in detecting regular patterns, and the null space function test is not very effective in detecting clustered patterns. Furthermore, from the expression of the function $L(r)$, in the K-Ripley function, it is evident that its minimum possible value is $L(r) = -r$. This minimum value coincides with the critical value estimated by the Monte Carlo simulations. Therefore, this test cannot detect a regular pattern since a pattern is considered regular if it is below the critical values estimated by the test. For $\widehat{G}$, it holds that $\widehat{G}(d) \geq 0$, for all d, the lower critical range estimated for the test of the distance to the nearest neighbor is $\widehat{G}(d) = 0$ 0. Therefore, this test will not be able to detect regular patterns either. Like the $\widehat{G}$ function, the minimum value that the $\widehat{F}$ function can take is 0. This minimum value coincides with the lower critical value estimated by Monte Carlo simulations. Therefore, this test is not capable of detecting clustered patterns. Of the 10,000 iterations of the F function test for the xy pattern, which expresses the greatest possible regularity between 5 points in a rectangle, it turns

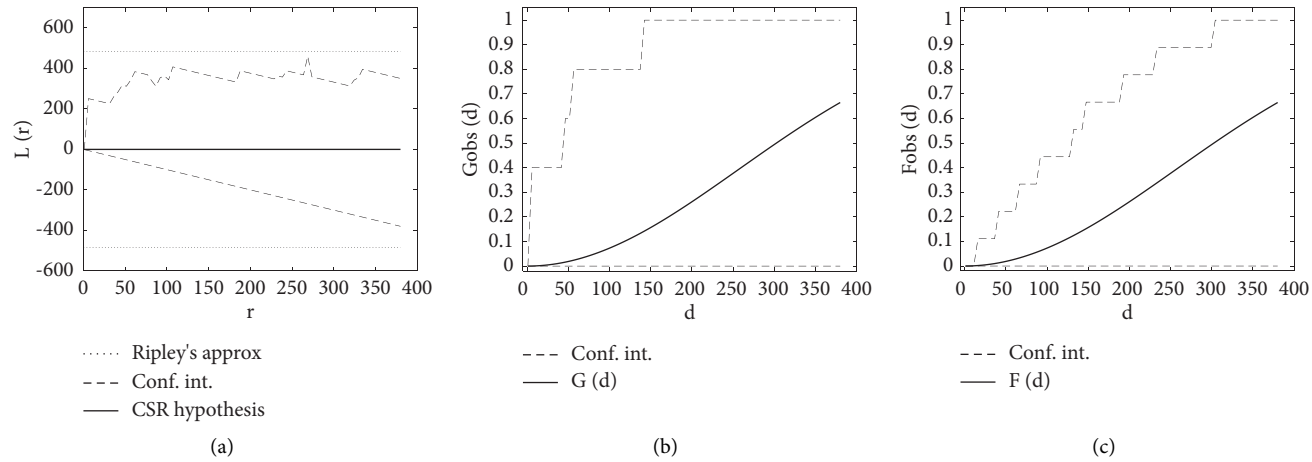


Figure 4: Critical values of the K-Ripley tests: (a) the nearest neighbor (b) and empty space (c) for 199 Monte Carlo simulations.

out that none of them detects said pattern as regular. These 10,000 iterations are because the F function depends on a grid, which is an additional set of random points; therefore, for a pattern, the value of the function can change depending on the grid. Then the 10,000 iterations were performed for the xy pattern but varying the grid so that the result did not depend on it.

The results obtained are summarized in Table 1, where t he sign "–" means that the corresponding test is not applicable in the case in question. The results show that the K-Ripley function and the nearest neighbor tests are not effective in detecting clustered 5-point patterns and are not capable of detecting regular 5-point patterns. For its part, the empty space distance test showed an effectiveness of 0% in detecting regular patterns and is unable to detect clustered patterns. Therefore, these three analyzed spatial randomness tests turn out to be ineffective in detecting nonrandom graphical passwords in the scenario PassPoint.

Recently, in [30], the application of the characteristic "number of sides of the Voronoi polygons" was evaluated for the detection of graphical passwords formed by patterns clustered in PassPoint, but it also proved to be ineffective using the proposed criteria.

3.2. The Sample Mean, Sample Variance, and Distribution of the Averages of the Perimeters of the Delaunay Triangles. In Section 2.4, we discussed the use of some of the features of Voronoi diagrams and Delaunay triangulations to detect spatial point patterns. In the PassPoint scenario, the points (pixels) of a clustered password are very close between them, and those of a regular graphical password are far from each other for a higher level of consistency. Considering this, in this work, we propose to use the perimeter of the Delaunay triangles to detect randomness between the password points instead of some other characteristic. However, it may be the case that in a password where the points are randomly distributed, the perimeter of one of its Delaunay triangles is just as small as that of one in a clustered password or just as big as one of the triangles of a password with regularly distributed points. In Figure 5, it is observed how the maximum perimeter of the Delaunay triangles of the clustered points coincides with the minimum perimeter of the Delaunay triangles of the random points, as the maximum perimeter of the triangles of Delaunay of the random points coincides with the minimum perimeter of the regular points. This suggests using the average of the perimeters of the Delaunay triangles as decision criteria to detect clustering or regularity between the pixels of a password in PassPoint and not the minimum or maximum value of the Delaunay triangles perimeter.

Table 1: Percentage of nonrandom graphical passwords detected by each test in each experiment.

Test	DB. $1.1_{Ag.(IV)}$	DB. $1.2_{Ag.(VIII)}$	xy
Null space	—	—	0%
K-Ripley	5.31%	26.55%	—
Nearest neighbor	1.88%	8.90%	—

Thus, it is then necessary to determine the probability distribution that best fits the distribution of the average of the perimeters of the Delaunay triangles of a password; for this, experiment 3 was designed and carried out in the following way. 1,000 random graphic passwords were generated in each of the three image sizes, 800×480, 1366×768, and 1920×1080 pixels, as the first image is the most common in mobile phones and the other two in computers. For each of these passwords, its Delaunay triangulation is constructed and the average of the perimeters of its Delaunay triangles is calculated, obtaining a total of three random databases of 1,000 averages each. The first database (DB.3.1) contains the averages of the image of 800×480 and the second one (DB.3.2) those of 1366×768, whereas the third one (DB.3.3) contains the averages of the last image. To measure the fit of the data to some known theoretical distribution, the EasyFit 5.6 software was used, which allows the distributions to be automatically adjusted to the sample data and the best model selected in a few seconds [40, 41]. The EasyFit 5.6 consists of 54 theoretical distributions, with some of them for various parameter sets, making a total of 61 possible options to fit for the data.

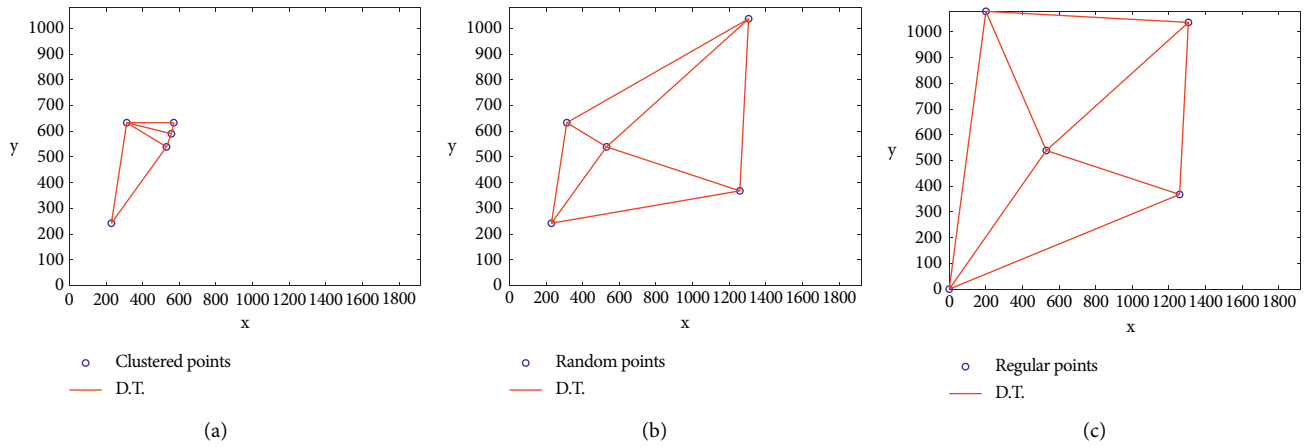


Figure 5: Delaunay triangulations of clustered points (a), random points (b), and regular points (c).

Table 2: Mean, sample variance, and standard deviation of the averages of the perimeters of Delaunay triangles (P_{P_D}) in the DBs: 3.1, 3.2, and 3.3, respectively.

P_{P_D}	DB.3.1	DB.3.2	DB.3.3
$E[P_{P_D}]$	872	1,439	2,038
$V[P_{P_D}]$	39,391	102,250	210,330
σ	199,902	319,766	458,617

Table 3: The six best theoretical distributions adjusted by the data from the random database (DB.3.1) with an image size of 800 × 480 pixels using the Kolmogorov–Smirnov (K-S), Anderson–Darling (A-D), and Chi-Square (χ^2), using the significance levels $\alpha \in \{0.2; 0.1; 0.05; 0.02; 0.01\}$, and its p-values associated with the Kolmogorov–Smirnov and Chi-Square tests.

Distribution	Number of acceptances	K–S	A-D	χ^2
Weibull (3P)	15/15	0.98070	Accepted	0.99858
Kumaraswamy	15/15	0.97874	Accepted	0.99892
Gen. Extreme value	5/10	0.94770	Rejected	N/A
Log-Pearson 3	15/15	0.94350	Accepted	0.98486
Johnson SB	15/15	0.94114	Accepted	0.99858
Weibull	15/15	0.77054	Accepted	0.65173

From experiment 3, we obtained the following results. Table 2 shows the sample mean and variance corresponding to the averages of the perimeters of the Delaunay triangles for each of the random password databases. Tables 3–5 show the six best models of distributions to which the data were fitted. Table 6 presents the results of the three goodness-of-fit tests applied to the Johnson SB distribution and the estimated distribution of the averages of the perimeters of the Delaunay triangles in each of the random databases corresponding to the sizes of studio images. However, when measuring the adjustment of the 1,000 averages of the perimeters of the Delaunay triangles estimated in each of the random databases to a known theoretical distribution, it was obtained that in each of the databases, it was possible to adjust the averages of the perimeters to more than 20 distributions, with some of them accepted by the three goodness-of-fit tests (Kolmogorov–Smirnov, Anderson–Darling, and Chi-square) with the significance levels $\alpha \in \{0.02, 0.01, 0.05, 0.1, 0.2\}$.

Table 4: The six best theoretical distributions adjusted by the data from the random database (DB.3.2) with an image size of 1366 × 768 pixels using the Kolmogorov–Smirnov (K–S), Anderson–Darling (A-D). and Chi-Square (χ^2), using the significance levels $\alpha \in \{0.2; 0.1; 0.05; 0.02; 0.01\}$, and its p-values associated with the Kolmogorov–Smirnov and Chi-Square tests.

Distribution	Number of acceptances	K–S	A-D	χ^2
Johnson SB	15/15	0.95628	Accepted	0.31172
Gen. Extreme value	5/10	0.95460	Rejected	N / A
Kumaraswamy	15/15	0.88420	Accepted	0.56845
Error	14/15	0.87086	Accepted	$\alpha \neq 0.2$ 0.16329
Weibull(3P)	15/15	0.79740	Accepted	0.50346
Log-Pearson 3	14/15	0.77399	Accepted	$\alpha \neq 0.2$ 0.15881

Table 5: The six best theoretical distributions adjusted by the data from the random database (DB.3.3) with an image size of 1920 × 1080 pixels using the Kolmogorov–Smirnov (K–S), Anderson–Darling (A-D), and Chi-Square (χ^2), using the significance levels $\alpha \in \{0.2; 0.1; 0.05; 0.02; 0.01\}$, and its p-values associated with the Kolmogorov–Smirnov and Chi-Square tests.

Distribution	Number of acceptances	K–S	A-D	χ^2
Error	15/15	0.99459	Accepted	0.69818
Johnson SB	15/15	0.98592	Accepted	0.83378
Gen. Extreme value	15/15	0.97157	Accepted	0.74561
Log-Pearson 3	15/15	0.96973	Accepted	0.73518
Kumaraswamy	15/15	0.90786	Accepted	0.66530
Weibull (3P)	15/15	0.90681	Accepted	0.55614

Table 6: Results of the three goodness-of-fit tests with the significance levels $\alpha \in \{0.02, 0.01, 0.05, 0.1, 0.2\}$, applied to the Johnson SB distribution estimated by the data for each of the random databases DBs: 3.1, 3.2, and 3.3.

Goodness-of-fit test	DB.3.1	DB.3.2	DB.3.3
Kolmogorov–Smirnov	0.94114	0.95628	0.98592
Chi-square	0.99858	0.31172	0.83378
Anderson–Darling	Accepted	Accepted	Accepted
Number of acceptances	15/15	15/15	15/15

Table 7: Parameters of the Johnson SB distribution $(\gamma, \delta, \lambda, \xi)$ of the averages of the perimeters of the Delaunay triangles in the DBs: 3.1, 3.2, and 3.3, respectively, $P_{P_D} \sim J_{SB}(\gamma, \delta, \lambda, \xi)$.

DB.	Image size	γ	δ	λ	ξ
DB.3.1	800×480	−0.44981	2.7884	2 295.3	−365.06
DB.3.2	1366×768	−0.25323	1.9873	2 700.2	8.2037
DB.3.3	1920×1080	−0.21458	2.0283	3 940.5	−30.961

We now discuss the results of experiment 3. Table 2 illustrates that the sample mean and variance differ between the databases due to the inequality between the image sizes. The averages of the perimeters of the Delaunay triangles belonging to the three sizes of the images under study did not fit the distributions with the same parameters (Table 7) or in the same order of the best models fitted by EasyFit, but the fitted distributions for each image size mostly match. Among the best distributions that fit the perimeters of the Delaunay triangles $(\forall \alpha)$ for the random databases DB.3.1, DB.3.2, and DB.3.3 is the Johnson SB, which occupies the fifth, first, and second place among the best possible models, respectively (Figure 6). This distribution allows for the transformation of the data to a standard normal distribution using the following formula [42]:

$$P_D^N = J_{SB}(P_D) = \gamma + \delta \times \ln\left[\frac{(P_D - \xi)}{(\lambda + \xi - P_D)}\right], \tag{10}$$

$P_D^N \sim N(0, 1)$. This transformation makes it easy to apply normality tests based on the fit of the data. Then, under the randomness hypothesis, the average of the perimeters of the Delaunay triangles of a graphical password in PassPoint when transforming the data to a standard normal distribution is 0. Therefore, it can be assumed that the passwords that violate the above proposition do not follow a random pattern.

3.3. Test Based on the Average of the Perimeters of the Delaunay Triangles. In this subsection, we propose a statistical test to detect nonrandom passwords in PassPoint. This test constitutes the main contribution of this article, considering that the classic tests are ineffective in detecting nonrandom graphical passwords in the PassPoint scenario. Although, recently [43], a test (of spatial randomness based on the mean distance between the points) was proposed with the same objective as the test proposed in this work, to detect nonrandom and, therefore, weak graphical passwords introduced by users during the registration phase in a PassPoint system, it is considered necessary to carry out in the next future works a comparison in terms of effectiveness and errors made between these two tests. The proposal of this work consists of a two-tailed hypothesis test based on the average of the Delaunay triangles' perimeters transformed to a standard normal distribution using the Johnson SB transformation. To apply this test, it is necessary to consider the size of the image selected by the user since the Johnson SB parameters are different for the sizes of images analyzed, as shown in Table 7.

3.3.1. Spatial Randomness Test Based on the Average of the Perimeter of Delaunay Triangles to Detect Nonrandom Passwords in PassPoint. We propose the following null hypothesis:

$$H_0\colon E\left[P_D^N\right] = E\left[J_{SB}(P_D)\right] = 0, \tag{11}$$

which states that the graphical password selected by the user is random if the average of the perimeters of the Delaunay triangles transformed by Johnson SB to a standard normal is equal to 0, with an alternative hypothesis given by H_1: $E(P_D^N) = E[J_{SB}(P_D)] \neq 0$. In order to test the hypothesis, the test statistic, based on the average perimeters of Delaunay triangles of the points of a user-selected password transformed by Johnson SB to a standard normal, is used. It is given by the following:

$$Z = J_{SB}\left(P_{P_D}\right) = \gamma + \delta \times \ln\left[\frac{\left(P_{P_D} - \xi\right)}{\left(\lambda + \xi - P_{P_D}\right)}\right]. \tag{12}$$

From Table 7, selecting the values of the transformation parameters depends on the image's size. The user or system can set the significance level α, whereas the critical region is $CR. = \{z\colon Z < -z_{\alpha/2} \text{ or } Z > z_{\alpha/2}\}$. Finally, with respect to the decision criteria, it is decided that the graphical password selected by the user does not follow a random pattern if, when transforming the average of the perimeters of its Delaunay triangles through the Johnson SB transformation, the value obtained belongs to the critical region.

3.4. Validation of the Effectiveness of the Proposed Test. To evaluate the effectiveness of the proposed test by means of type I and type II errors, Experiments 4 and 5 were carried out, respectively.

To estimate the probabilities of type I error from the proposed decision criterion, experiment 4 was designed. Three new random databases were generated, DB.4.1, DB.4.2, and DB.4.3, with 10,000 random graphical passwords each in each of the three image sizes, 800×480, 1366×768, and 1920×1080 pixels, respectively.

The results of experiment 4 are shown in Table 8. Note that the probability of obtaining the type I error corresponds approximately to the established level of significance (alpha theoretical) for all cases, which shows that the probabilities

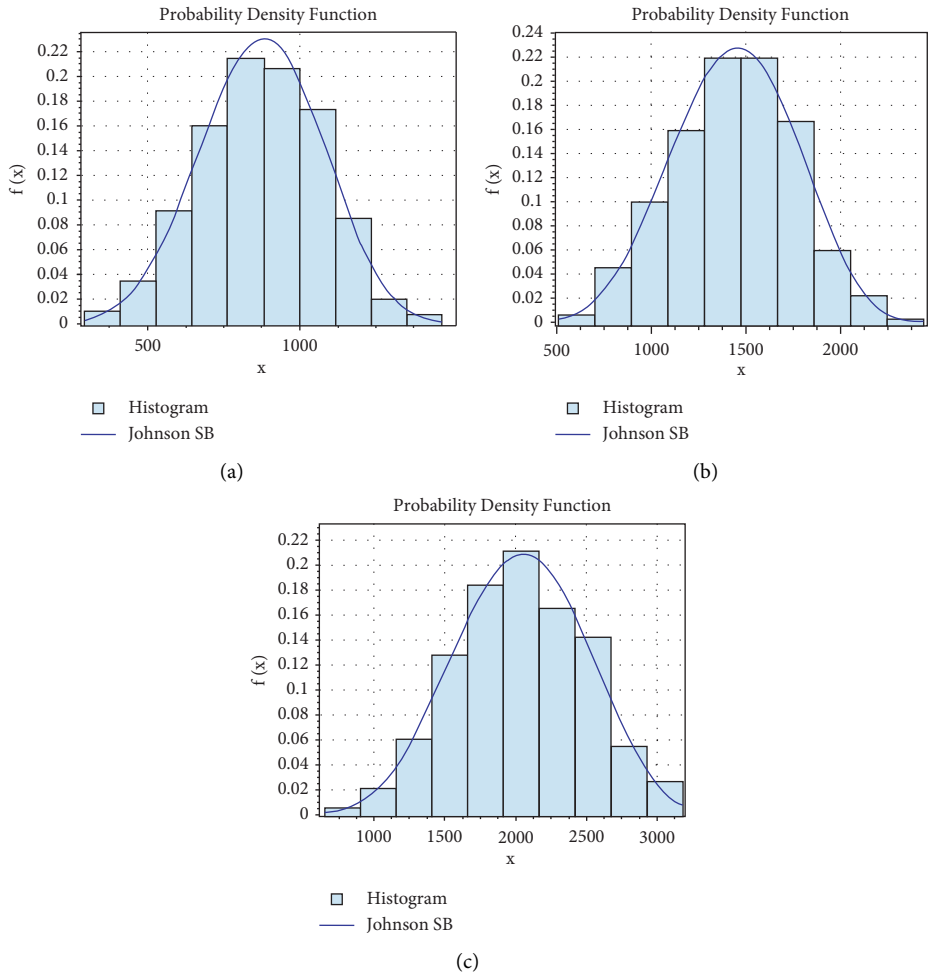


Figure 6: Histograms of the averages of the perimeters of the Delaunay triangles associated with the random graphical passwords in DB.3.1 (a), DB.3.2 (b), and DB.3.3 (c), respectively, and their comparison with the Johnson SB.

Table 8: Estimation of type I error (estimated alpha, $\widehat{\alpha}$), that is, of the probability that in DB.4.1, DB.4.2, and DB.4.3, the average of the perimeters of the triangles of a random graphical password belongs to the critical region. Comparison with the preset theoretical alpha (α).

α (Theoretical)	$CR.$ Of H_0	$\widehat{\alpha}_1$ DB.4.1	$\widehat{\alpha}_2$ DB.4.2	$\widehat{\alpha}_3$ DB.4.3	$\sum_{i=1}^{3} \widehat{\alpha}_i/3$
0.2	$Z < -1.282 o Z > 1.282$	0.1803	0.1962	0.1885	0.1883
0.1	$Z < -1.645 o Z > 1.645$	0.0853	0.1023	0.0918	0.0931
0.05	$Z < -1.960\, o\, Z > 1.960$	0.0403	0.0545	0.0499	0.0482
0.02	$Z < -2.326 o Z > 2.326$	0.0166	0.0248	0.0213	0.0209
0.01	$Z < -2.575 o Z > 2.575$	0.0081	0.0157	0.0116	0.0118

of type I errors do not seem to depend on the image size and that the proposed decision criterion is valid.

Now, for experiment 5, 50,000 nonrandom graphical passwords are generated in total, 30,000 clustered (10,000 in an area equivalent to a quarter of the image, 10,000 in an area equal to one-sixth of the image, and the other 10,000 in an area equivalent to the eighth of the image), and regular 20,000 (with a lower and higher level of regularity), for each of the study images. This means that, for the 800×480 image, the aggregation distances were $175u$, $145u$, and $125u$ radius; for the 1366×768, they were $290u$, $240u$, and $210u$ of radius; for the image of 1920×1080, the aggregation distances were $410u$, $335u$, and $290u$ of radius, respectively; the regular databases were generated by inhibition distances of $140u$ and $220u$, $210u$ and $350u$, and $300u$ and $505u$ of radius, respectively. The regular patterns were derived from a simple inhibition process: random locations of points were generated, verifying that at each new point, the distance to its closest neighbor was equal to or greater than a specified inhibition distance [8, 37]. In each of these databases, the type II error was estimated, and the number of passwords detected was calculated for the different levels of clustering and regularity.

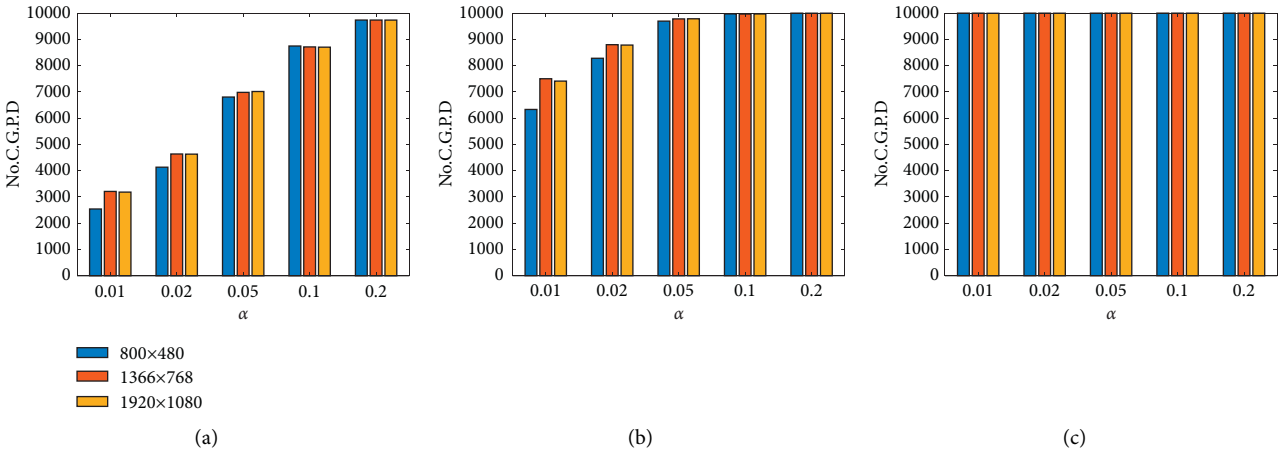


Figure 7: Number of clustered graphical passwords detected (No. of CGPD) in each of the image sizes for clustered pattern databases (in an area equivalent to one-fourth (a), one-sixth (b), and one-eighth (c) of the image), with significance level α.

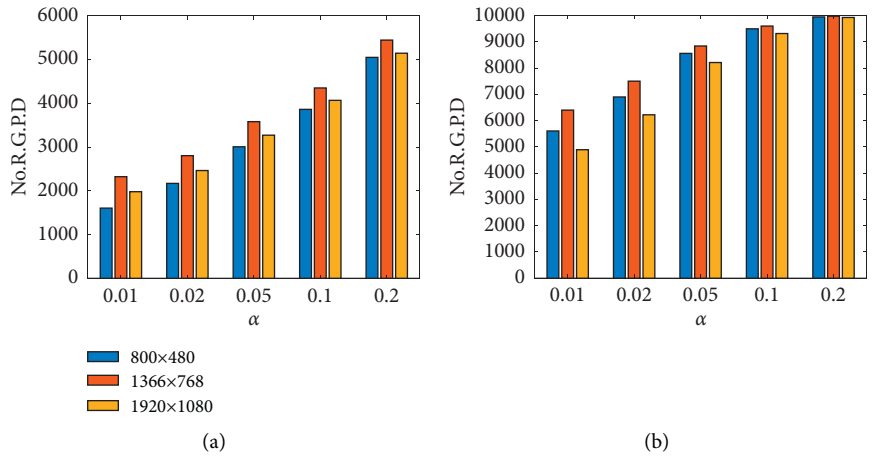


Figure 8: Number of regular graphical passwords detected (No. of RGPD) in each of the image sizes for databases with regular patterns (with lower (a) and higher (b) levels of regularity), with significance level α.

The results of experiment 5 are as follows. Figures 7 and 8 show the number of nonrandom graphical passwords detected in each of the nonrandom databases for the analyzed image sizes, and Table 9 represents the probabilities of type II errors estimated in nonrandom databases for an image size of 1920×1080.

These results clearly show that by increasing the level of clustering or the regularity level, the test becomes more effective, as was to be expected. The decision criterion is usually quite effective in detecting clustered graphical passwords, especially for the significance levels $\alpha = 0.1$ and $\alpha = 0.2$ for which it detects 87% and 97% of the passwords, respectively (see Figure 7 and Table 9), in an area equivalent to one-fourth of the image; on the other hand, in the regular graphical passwords with a lower level of regularity, for $\alpha = 0.2$, it only detects approximately 50 of the passwords (see Figure 8 and Table 9). The criterion reaffirms Chiu's approach in [9], since the average of the Delaunay triangles' perimeters is more effective in detecting clustering than regularity. Figures 7 and 8 show that the probabilities of type II errors do not seem to depend on the image size since their estimated values are similar for the different sizes of images; therefore, only the type II error was shown (Table 9) for each of the nonrandomized study databases of one of the image sizes.

This test was designed exclusively to detect graphical passwords with clustered or regular patterns in PassPoint. Therefore, other types of patterns identified in the bibliography [22], such as soft ones or with different predetermined shapes (see Figure 9), will only be detected by the test proposed if these also present a certain level of clustering or regularity (as shown in Figure 10). Therefore, if the patterns are not clustered, it cannot be said that the test can detect these patterns since these patterns have to fulfill the property that when forming their respective Delaunay triangles, one of the interior angles of the triangle has to be obtuse so that the triangle is as devoid of peaks as possible and a relatively smooth curve is formed. Visually, it could be interpreted as patterns in the form of a straight line (or almost straight, given the

Table 9: Probability estimated $(\widehat{\beta})$ in DB. $5.1.1_{Ag.(IV)}$, DB. $5.1.2_{Ag.(VI)}$, DB. $5.1.3_{Ag.(VIII)}$, DB. $5.2.1_{\text{Reg}}$ (less regular), and DB. $5.2.2_{+_{\text{Reg}}}$ (more regular) to accept a random graphical password when it is actually a clustered or regular graphical password.

Significance Level	Error of Tipo II	$(\widehat{\beta})$ DB. $5.1.1_{Ag.(IV)}$	$(\widehat{\beta})$ DB. $5.1.2_{Ag.(VI)}$	$(\widehat{\beta})$ DB. $5.1.3_{Ag.(VIII)}$	$(\widehat{\beta})$ DB. $5.2.1_{\text{Reg.}}$	$(\widehat{\beta})$ DB. $5.2.2_{+_{\text{Reg}}}$
0.2	$-1.282 < Z < 1.282$	0.0262	0.0002	0	0.4856	0.0047
0.1	$-1.645 < Z < 1.645$	0.1293	0.0034	0	0.5933	0.0500
0.05	$-1.960 < Z < 1.960$	0.2982	0.0219	0	0.6729	0.1436
0.02	$-2.326 < Z < 2.326$	0.5368	0.1201	0.0001	0.7537	0.3096
0.01	$-2.575 < Z < 2.575$	0.6814	0.2498	0.0005	0.8021	0.4391

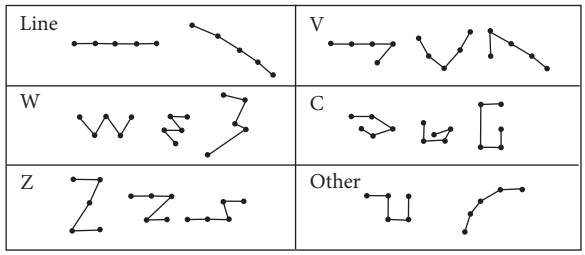
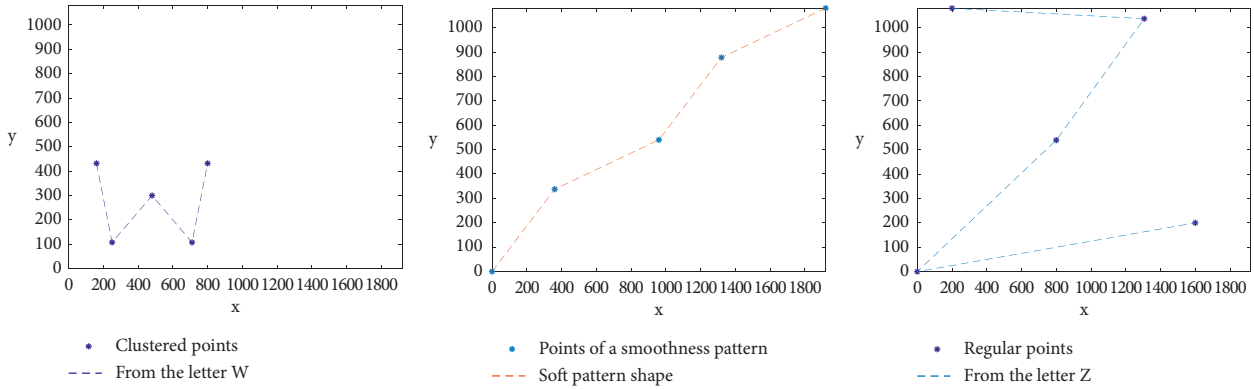


Figure 9: Patterns with different predetermined shapes.



(a) (b) (c)

Figure 10: Pattern with default shape w, which also follows a clustering pattern (a), the pattern on with default shape (soft) but is detected as random (b), and pattern on with default shape Z, which also follows a regularity pattern (c).

Table 10: Number and proportion of nonrandom graphical passwords detected in the databases DB. $1.1_{Ag.(IV)}$, DB. $1.2_{Ag.(VIII)}$, and the regularity pattern xy, by means of the empty space function, the nearest neighbor distance test, the K-Ripley function, and the proposed test.

Test	DB. $1.1_{Ag.(IV)}$	DB. $1.2_{Ag.(VIII)}$	xy
Empty space	–	–	0%
Nearest neighbor	$188/10,000 = 0.0188$	$890/10,000 = 0.0890$	–
K-Ripley	$531/10,000 = 0.0513$	$2655/10,000 = 0.2655$	–
Proposed test	$3,212/10,000 = 0.3212$	$10,000/10,000 = 1.0000$	Detected

low probability that the user will select the points of his graphical password in such a way that they form exactly a straight line). This discussion suggests that a test to detect weak passwords can be constructed from the Delaunay triangles' interior angles, which is left proposed for future work, as well as its comparison with the test proposed in [44].

3.5. Comparison in PassPoint of the Proposed Test and the Tests Most Used in CSR. Table 10 shows the comparison between the proposed test, the K-Ripley function, the test of the distance to the nearest neighbor, and the empty space function in terms of the effectiveness in the detection of clustered and regular graphical passwords onstage PassPoint, for a significance level of $\alpha = 0.01$.

The image size of 1920 × 1080 pixels was used to make this comparison. The results for the other sizes of images studied in this work have a similar behavior. For an image of this size, the average of the perimeters of the Delaunay triangles of the pattern xy is $3,702.9u$, whereby transforming this average from a Johnson SB distribution to a standard normal using the statistic Z (12) to get $Z = 5.6558 > 2.575 = z_{0.005}$. Then, by means of the proposed test, the xy pattern is rejected with a 99 confidence, the expected occurrence given its ability to detect regular graphical passwords. This convincingly demonstrates the superiority of the proposed test over the classical tests of spatial randomness to detect nonrandom passwords in PassPoint.

3.6. Application of the Proposed Test in PassPoint. In graphical authentication, in the PassPoint scenario, the proposed spatial randomness test allows the user to verify the strength of their password during the registration phase. This is possible due to its ability to detect spatial patterns of clustering or regularity between the points that make up the password. The user must define the level of significance with which they want to verify their password, although it is recommended to use $\alpha = 0.2$ for greater effectiveness. During the PassPoint registration phase, the test can be included by following these steps:

Step 1. The user selects the 5 points (pixels) of his password in an image.

Step 2. Calculate the average of the perimeters of the Delaunay triangles in the password.

Step 3. Calculate the test statistic Z Equation (11) by performing the Johnson SB transformation to the average of the perimeters calculated in Step 2.

Step 4. Determine the critical region taking into account the specified significance level.

Step 5. Decision criteria: if the test statistic calculated in Step 3 does not belong to the critical region, the registration is successfully completed, but if it belongs to the critical region, the user is notified that the password is weak and returns to Step 1.

The proposed test must apply to other systems of the cued-recall type that uses 5 points, or a number close to 5, as its graphical password in an image. The experiments that prove it are left to be published in future research.

4. Conclusions and Future Work

In this work, it was shown that three of the most used classical tests in complete spatial randomness are inefficient in detecting nonrandom passwords in the PassPoint scenario, so the average of the perimeters of the Delaunay triangles was investigated to extract dependency information between password points. Its distribution was estimated in each of the random databases, which was adjusted to more than 20 known distributions for each of the study image sizes, the Johnson SB distribution for each image being among the five best fits. Different parameters of the Johnson SB distribution were obtained from the averages of the perimeters of the Delaunay triangles for the three sizes of images analyzed. Therefore, it was assumed with an established significance level that graphical passwords that violate this property are not random. The application of this criterion is facilitated because after applying the Johnson SB transformation with the parameters of the Johnson SB distribution established for each image size, the transformed average must follow a standard normal distribution. Based on the average of the Delaunay triangles perimeters transformed to a standard normal distribution by the Johnson SB transformation, a test was proposed to detect weak graphical passwords formed by clustered or regular points. Type I and type II errors were estimated, and the number of graphical passwords detected by this test was calculated for various levels of clustering and regularity. It was concluded that regardless of the image size, their estimates of type I and type II errors roughly coincide for an established level of significance and thus, the number of passwords detected. It is concluded that the proposed criterion based on the average of the perimeters of the Delaunay triangles is efficient for detecting weak graphical passwords in PassPoint, formed by five clustered points or by five regular points, although it is more precise in detecting clustering than regularity. Despite the effectiveness of the proposed test being tested for various levels of clustering or regularity, with different type II errors, the minimum level of clustering or regularity for which the test's effectiveness remains acceptable in application practices is still unknown. This aspect will be investigated in future work. Another open problem that will be discussed soon is the reduction of type II errors. The proposed 2-tailed test assesses deviations from randomness, and its effectiveness was evaluated in the detection of two types of patterns, clustered or regular. If hypotheses of the type H_1: clustered or H_1: regular are considered separately as alternative hypotheses, a one-tailed test will be obtained in each case, and a reduction of the type II error can be expected. This approach has the limitation of evaluating the existence of a specific type of nonrandom pattern, and a different test should be applied for each type of pattern. Its advantage is that it can be more effective in determining the type of pattern once it is decided to reject randomness. In future works, experiments will be carried out to evaluate the proposed test to detect passwords formed by soft patterns or with different predetermined forms. Another aspect to evaluate is the comparison in terms of effectiveness and errors made of the proposed test and the spatial randomness test based on the mean distance between the points. In addition, combinations of the different tests will be analyzed to increase the effectiveness in detecting nonrandom passwords without significantly compromising the implementation time. It is also proposed to evaluate the effectiveness of other characteristics of Delaunay triangulation to detect patterns in PassPoint, such as the minimum angle of a Delaunay triangle to detect regularity

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Swapnil Sunil, D. Prakash, and Y. Ramesh Shivaji, "Cued click points: graphical password authentication technique for security," *International Journal of ComputerScience and Information Technologies*, vol. 5, no. 2, 2014.

[2] O. Rodriguez, C. M. Legón, and R. Socorro, "Seguridad y usabilidad de los esquemas y técnicas de autenticación gráfica," *Revista Cubana de Ciencias Informáticas*, vol. 12, no. 13–27, 2018.

[3] S. Wiedenbeck, J. Waters, J. C. Birget, A. Memon, and N. Memon, "PassPoints: design and longitudinal evaluation of a graphical password system," *International Journal of Human-Computer Studies*, vol. 63, no. 1-2, pp. 102–127, 2005.

[4] H. Gao, W. Jia, F. Ye, and L. Ma, "A survey on the use of graphical passwords in security," *Journal of Software*, vol. 8, no. 7, pp. 1678–1698, 2013.

[5] R. G. Rittenhouse, J. Ahsenali Chaudry, and M. Lee, "Security in graphical authentication," *International Journal of Security and Its Applications*, vol. 7, no. 3, pp. 347–356, 2013.

[6] O. Rogriguez, C. M. Legón, R. Socorro, and P. Navarro, "Patrones en el orden de los clics y su influencia en la debilidad de las claves en la técnica de autenticacion gráfica passpoints," *Revista Cubana de Ciencias Informáticas*, vol. 12, no. 7, pp. 37–47, 2019.

[7] P. C. van Oorschot and J. Thorpe, "Exploiting predictability in click-based graphical passwords," *Journal of Computer Security*, vol. 19, no. 4, pp. 669–702, 2011.

[8] A. Baddeley, E. Rubak, and R. Turner, *Spatial Point Patterns: Methodology and Applications with R*, CRC Press, Boca Raton, FA, USA, 2015.

[9] S. N. Chiu, "Spatial point pattern analysis by using Voronoi diagrams and Delaunay tessellations - a comparative study," *Biometrical Journal*, vol. 45, no. 3, pp. 367–376, 2003.

[10] A. E. Gelfand, P. Diggle, P. Guttorp, and M. Fuentes, *Handbook of Spatial Statistics*, Handb. Spat. Stat. Chapman & Hall/CRC, Boca Raton, FA, USA, 2010.

[11] B. Li, Q. Meng, and H. Holstein, "Point pattern matching and applications - A review," in *Proceedings of the SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, vol. 1, IEEE, Washington, DC, USA, November 2003.

[12] N. Oliver and D. Knitter, *Modelling human behaviour in landscapes*, Springer International Publishing, New York, NY, USA, 2016.

[13] E. A. Wentz, "Pattern analysis based on type, orientation, size, and shape," *Geographical Analysis*, vol. 40, no. 2, pp. 97–122, 2008.

[14] M. Erwig, "The graph voronoi diagram with applications," *Networks*, vol. 36, no. 3, pp. 156–163, 2000.

[15] B. B. Zhu, D. Wei, M. Yang, and J. Yan, "Security implications of password discretization for click-based graphical passwords," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 1581–1591, Association for Computing Machinery, Rio de Janeiro, Brazil, May 2013.

[16] J. C. Birget, D. Hong, and N. Memon, "Robust Discretization, with an Application to Graphical Passwords," 2004, https://eprint.iacr.org/2003/168.

[17] S. Chiasson, J. Srinivasan, R. Biddle, and P. C. van Oorschot, "Centered discretization with application to graphical passwords (full paper)," in *Proceedings of the UPSEC'08: Proceedings of the 1st Conference on Usability, Psychology, and Security*, Berkeley, CA, USA, April 2008.

[18] K. Bicakci, "Optimal discretization for high-entropy graphical passwords," *1008 23rd Int. Symp. Comput. Inf. Sci. Isc.*vol. 2008, 2008.

[19] D. Kirovski, N. Jojić, and P. Roberts, "Click passwords," *IFIP Int. Fed. Inf. Process.*vol. 201, pp. 351–363, 2006.

[20] A. Emir Dirik, N. Memon, and J. Camille Birget, "Modeling user choice in the passpoints graphical password scheme," in *Proceedings of the 3rd symposium on Usable privacy and security*, vol. 20–28, Pittsburgh, PA, USA, July 2007.

[21] M. Devlin, R. Nurse, C. Hodges, H. Goldsmith, M. Creese, and S. Creese, "Predicting graphical passwords," in *Proceedings of the International Conference on Human Aspects of Information Security, Privacy and Trust at the 17th International Conference on Human-Computer Interaction (HCI)*, pp. 23–35, Springer International Publishing, Los Angeles, CA, USA, August 2015.

[22] S. Chiasson, A. Forget, R. van Oorschot, and P. C. Van Oorschot, "User interface design affects security: patterns in click-based graphical passwords," *International Journal of Information Security*, vol. 8, no. 6, pp. 387–398, 2009.

[23] P. C. van Oorschot, A. Salehi-Abari, and J. Thorpe, "Purely automated attacks on PassPoints-style graphical passwords," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 393–405, 2010.

[24] J. D. Peter, *Statistical Analysis of Spatial and Spatio-Temporal point Patterns*, CRC Press, Boca Raton, FA, 2013.

[25] J. Illian, A. Penttinen, H. Stoyan, and S. Dietrich, *Statistical analysis and modelling of spatial point patterns*, Wiley, Hoboken, NJ, USA, pp. 1–534, 2008.

[26] Y. Beatriz Caballero and H. Giraldo Ramón, "Test de aleatoriedad para procesos puntuales espaciales basados en el cálculo de la dimensión fractal. Tesis presentada para optar el título de Magíster en Ciencias Estadísticas," Technical report, Universidad Nacional de Colombia, Bogotá, Colombia, 2017.

[27] M. De La Cruz, "Métodos para analizar datos puntuales," *Introd. al Análisis Espac. Datos en Ecol. y Ciencias Ambient. Métodos y Apl*, pp. 75–127, Asociación Española De Ecología Terrestre, Madrid, Spain, 2008.

[28] T. Wiegand and A. Kirk, *Handbook of spatial point-pattern analysis in ecology*, Routledge, England, UK, 2013.

[29] A. Okabe, B. Boots, K. Sugihara, S. N. Chiu, and D. G. Kendall, "Definitions and basic properties of voronoi diagrams," *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, Vol. 43–112, John Wiley & Sons, , Hoboken, NJ, USA, 2000.

[30] L. Suárez-Plasencia, J. A. Herrera-Macías, and C. M. Legón-Pérez, "Analysis of the number of sides of voronoi polygons in passpoint," in *Proceedings of the Computer Science and Health Engineering in Health Services: 4th EAI International Conference, COMPSE 2020*, vol. 4, pp. 184–200, Springer International Publishing, New York, NY, USA, November 2021.

[31] B. N. Boots, "Using angular properties of delaunay triangles to evaluate point patterns," *Geographical Analysis*, vol. 18, no. 3, pp. 252–259, 2010.

[32] M. Deng, Q. Liu, T. Shi, and Y. Shi, "An adaptive spatial clustering algorithm based on delaunay triangulation," *Computers, Environment and Urban Systems*, vol. 35, no. 4, pp. 320–332, 2011.

[33] R. Bembenik, A. Protaziuk, and G. Protaziuk, "Discovering collocation rules and spatial association rules in spatial data with extended objects using Delaunay diagrams," in *Rough Sets and Intelligent Systems Paradigms*, vol. 8537 LNAI, pp. 293–300, Springer-Verlag, Berlin, Germany, 2014.

[34] R. Bembenik, W. Protaziuk, and G. Protaziuk, "Methods for mining co-location patterns with extended spatial objects," *International Journal of Applied Mathematics and Computer Science*, vol. 27, no. 4, pp. 681–695, 2017.

[35] B. Calka, E. Bielecka, and M. Figurski Open Geosciences, *Spatial Pattern of ASG-EUPOS Sites*, Degruyter.com, Berlin, Germany, 2017.

[36] L. Andronov, J. Michalon, and K. Ouararhni, "3D Clustering Analysis of Super-resolution Microscopy Data by 3D Voronoi Tessellations," *Bioinformatics*, vol. 34, 2017.

[37] V. Camarero and J. J. Camarero, "Spatial analysis techniques applied in forest ecology: point pattern analyses," *Investigación Agraria: Sistemas y Recursos Forestales*, vol. 14, no. 1, p. 79, 2005.

[38] J. A. Herrera-Macías, L. Suárez-Plasencia, and C. M. Legón-Pérez, "Effectiveness of some tests of spatial randomness in the detection of weak graphical passwords in passpoint," in *Proceedings of the Computer Science and Health Engineering in Health Services: 4th EAI International Conference, COMPSE 2020*, vol. 4, pp. 173–183, Springer International Publishing, New York, NY, USA, November 2021.

[39] B. D. Ripley, "Tests of 'randomness' for spatial point patterns," *Journal of the Royal Statistical Society: Series B*, vol. 41, no. 3, pp. 368–374, jul 1979.

[40] K. Schittkowski, *Numerical data fitting in dynamical systems: a practical introduction with applications and software*, Vol. 77, Springer Science & Business Media, , Berlin, Germany, 2002.

[41] K. Schittkowski, "Easy-Fit: A Software System for Data Fitting in Dynamical Systems," *Structural and Multidisciplinary Optimization*, vol. 23, 2002.

[42] I. Juliana Lagos and J. A. Vargas, "Sistema de familias de distribuciones de Johnson, una alternativa para el manejo de datos no normales en cartas de control," *Revista Colombiana de Estadística*, vol. 26, no. 1, 2003.

[43] J. A. Herrera-Macías, C. M. Legón-Pérez, L. Suárez-Plasencia, L. R. Piñeiro-Díaz, O. Rojas, and G. Sosa-Gómez, "Test for detection of weak graphic passwords in passpoint based on the mean distance between points," *Symmetry*, vol. 13, no. 5, p. 777, 2021.

[44] O. Rodriguez, "Algoritmo para la detección de contraseñas gráficas con patrón de suavidad en la técnica de autenticación gráfica passpoints. tesis presentada en opción al título máster en ciencias matemáticas," Technical report, Universidad De La Habana, Havana, Cuba, 2019.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5682891, 11 pages
https://doi.org/10.1155/2022/5682891

Research Article

A Three-Stage Alternative Optimization Promoting Multi-UAV-Assisted Mobile Offloading

Xiao Han, Huiqiang Wang, Guangsheng Feng, Xiaoxiao Zhuang, and Chengbo Wang

College of Computer Science and Technology Harbin Engineering University, Harbin 115001, China

Correspondence should be addressed to Huiqiang Wang; wanghuiqiang@hrbeu.edu.cn

Received 18 January 2022; Revised 27 May 2022; Accepted 9 July 2022; Published 23 August 2022

Academic Editor: Zengpeng Li

A solution using multiple-relay method is presented to provide an efficient solution using UAV-assisted mobile devices to complete computation offloading tasks, which considers the UAVs in the offloading system to help mobile devices to offload tasks in remote areas. Additionally, a mixed integer nonlinear optimization (MINO) problem is constructed to maximize minimum user computational speed, and a three-stage iterative optimization algorithm is proposed to find a solution of the MINO problem. Simulation experiments are setup to verify the effectiveness of the devised methods, which show that our proposed algorithm and solution is superior to the single UAV method.

1. Introduction

With the blooming of Internet of things (IoT) and wireless mobile networks, smart mobile terminals are extensively considered in high speed information transmission systems to provide powerful platform for various intelligent applications, such as interactive games, augmented/visual realities (ARs/VRs), and unmanned driving, and so on. However, computation resource and battery budget limitations are main obstacle for achieving higher performance. As we know, mobile edge computing, which is a promising paradigm and a new technique to enhance computation speed and robust information transmissions and sharing, uses cloud servers at network edges. Then, one of the effective methods is to offload the data and computation tasks to servers to improve the system performance. For another case, there is little available infrastructures in use, such as disaster scenarios, military maneuver, and so on. Fortunately, unmanned aerial vehicles (UAVs) has been used and developed for assisting mobile edge computing (MEC) to tackle these challenges with less infrastructures.

In UAV-assisted MEC networks, computation and communication resources are optimized for achieving objective system design, including minimization consumption energy [1, 2] and minimization system cost [3]. Then, the energy reduction problem is investigated in UAV-enhanced edge via smart offloading decisions and allocating transmitting bits in both uplink and downlink [1]. The energy optimization problem has been extended into multi-UAV-assisted MEC systems using an iterative algorithm with double-loop structure to find optimal solution. To minimize the system cost of vehicle computing tasks, a software defined network (SDN)-derived UAV-assisted vehicular computation offloading optimization framework has been reported to construct a multiplayers offloading sequential game [3]. However, these works focused on optimizing a single objective [1]. After that, multiobjective optimization problems for UAV-assisted MEC network have been presented [4, 5] to provide a balance between the CPU frequencies, the offloading amount, the transmit power, and the UAV' s trajectory. Additionally, the mission completion time was minimized though jointly optimizing UAV trajectory and communication resource allocation [6] and air base-stations (BSs) with multiple UAVs are used to provide services for users on the ground using multiple UAVs in a wireless communication [7], where UAV trajectory and power can be controlled by optimizing multiuser communication scheduling and association, resulting in

maximization throughput for all terrestrial users in downlink communications and hence achieving fair performance between users. Then, a UAV interference channel (UAV-IC) is considered in which each UAV communicates with the associated ground terminal (GT) on the same spectrum to establish a joint trajectory and power control (TPC) to maximize the total power of UAV-IC over a given flight interval [8]. However, all the ideas consider the densely deployed scenarios, which cannot always hold when the BSs are damaged by natural disasters. In addition, these works considered the single demand of users, while both uploading and downloading requirements of users have not been well studied.

In contrast to [1–10], the resource allocation problem for multi-UAV-assisted MEC system is investigated and studied, where we consider different offloading requirements for each user. In the devised system, multi-UAV promoting MEC system can be used for mountain and desert damage senses. In the proposed scheme, the UAVs should fulfill offloading before decision and avoid collisions, in which the offloading decision, resources allocation and UAV's trajectory planning are jointly considered to find an efficient solution using multi-UAV-assisted mobile devices to complete computation offloading tasks. Also, a mixed integer nonlinear optimization (MINO) problem is solved by maximizing minimum user computational speed. In addition, a three-stage iterative optimization algorithm is proposed to find a solution of the MINO problem that is a NP-hard problem. Simulation experiments are setup to verify the effectiveness of the devised methods, which show that our proposed algorithm and solution is superior to the single UAV method. The main contributions of this paper are summarized as follows.

(1) Considering both user data offloading and computation offloading, the multiple UAVs-assisted mobile offloading (MUMO) problem is formulated by considering maximization minimum user calculation rate.

(2) The MUMO problem is carefully considered and divided into three sub-problems, namely, resource allocation, trajectory optimization and anti-collision and offloading decision. The closed form of optimal solution for resource allocation is obtained and analyzed in detail.

(3) A three-stage iterative optimization (TSIO) algorithm is proposed to solve the three sub-questions given above based on successive convex approximation (SCA) methods.

As mentioned above, a solution using multiple-relay method is presented to provide an efficient solution using UAV-assisted mobile devices to complete computation offloading tasks, which considers the UAVs in the offloading system to help mobile devices to offload tasks in remote areas. Additionally, a mixed integer nonlinear optimization (MINO) problem is constructed to maximize minimum user computational speed, and a three-stage iterative optimization algorithm is proposed to find a solution of the MINO problem. Simulation experiments are setup to verify the effectiveness of the devised methods, which show that our proposed algorithm and solution is superior to the single UAV method.

In this paper, we proposed TSIO algorithm to address the multi-UAV-assisted mobile computation offloading and the simulation experiments are constructed to verify the performance of the proposed scheme and TSIO algorithm. The rest of this work is organized as follows. Section 2 introduces the offloading model and presents the optimization problem. The property of the MUMO problem and the TSIO algorithm are proposed in Section 3. The numerical results and the conclusions are given in Sections 4 and 5, respectively.

2. Multi-UAV-Assisting Mobile Offloading Model

Considering a multi-UAV-assisted mobile computing and offloading scenario, each UAV can serve multiple users, while one user can only select one UAV. In this case, users are divided in three types, namely, users with computation offloading demand, users with traffic offloading demand, and users with both offloading demands. Let $N = 1, 2, \ldots, N$ denote the set of users and $K = 1, 2, \ldots K$ denote the set of UAVs. The position coordinates of user i can be expressed as $\mathbf{L}_i = [x_i^l, y_i^l, 0]$, where x_i^l represents the horizontal coordinate of the user location and y_i^l is their ordinate values. All users are fixed on the ground [11–13], and all UAVs fly in the same plane and have a fixed starting point q_{st} and ending point q_{end}. Moreover, the UAV flies at a fixed altitude, which is the height that guarantees normal flying and does not encounter obstacles, and can guarantee normal communication with all users. The wireless channel mode between UAVs and each user adopts the LOS [14] mode, and the channel loss model is Path loss model. The maximum flight speed of the UAV is $v_{\max}$ [15], while the total time from the start point to end point is T that is divided into M slots. Then, the UAV position at each slot can be expressed as $\mathbf{q}_j(t) = [x_j^{(c)}(t), y_j^{(c)}(t), H]$. Assume that there is no interfere between uplink and downlink and the bandwidth is BHz. In addition, the uplink and downlink adopt time division multiplexing (TDM) technology. The specific scenario is shown in Figure 1. Each UAV has its own trajectory, and each user can only select one UAV for service. Moreover, User 2 and User 4 have both uploading and downloading requirements, while User 1 has only downloading requirements, and User 4 has only uploading requirements. User 1 selects UAV2 for service, which may be due to excessive load on UAV2, also including distance and other factors. Therefore, the matching between user and UAV is mainly determined UAV load and UAV distance, UAV power, and so on. $b_{ij}(t)$ represents whether user i chooses UAV j for service at slot t. Herein, $b_{ij}(t)$ is a binary variable. When $b_{ij}(t)$ equals to 1, it means that user chooses the UAV for serving, and vice versa.

The frequency division duplexing (FDD) mode with equal channel bandwidth is adopted for both uplink and downlink. $\alpha_i^{(u)}(t)$ and $\alpha_i^{(d)}(t)$ represent the proportion of upload and download time allocated to users, respectively. $\theta_i^{(u)}$ indicates whether user has an upload request, and $\theta_i^{(d)}$ indicates whether user has a download request.

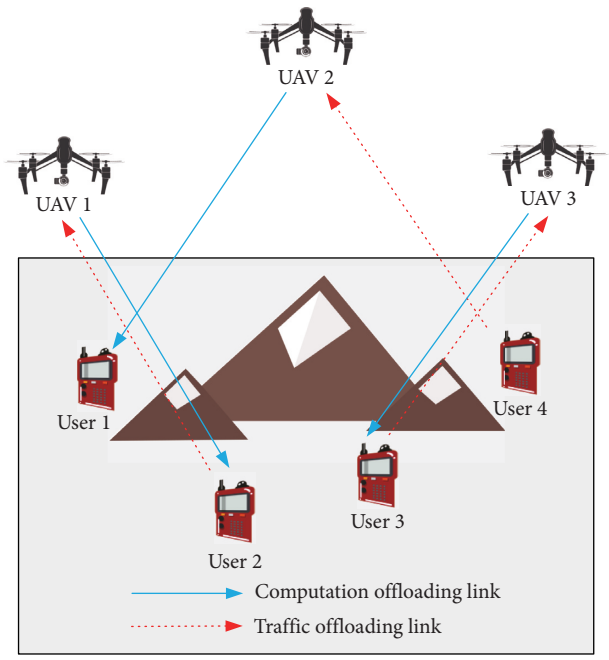


Figure 1: Multi-UAV assisting users with mobile offloading.

According to TDM, the sum of the upload time ratios is constrained by

$$\sum_{i=0}^{N} \theta_i^{(u)} \alpha_i^{(u)}(t) \le 1, \tag{1}$$

while the sum of download time ratios is bounded by

$$\sum_{i=0}^{N} \theta_i^{(d)} \alpha_i^{(d)}(t) \le 1, \tag{2}$$

where $\theta_i^{(u)}$, $\theta_i^{(d)}$, $\alpha_i^{(u)}(t)$, and $\alpha_i^{(d)}(t)$ are notified to UAV in advance. Here, the user is restricted to select only one UAV for service in each time slot, and hence, we have

$$\sum_{j=1}^{K} b_{ij}(t) = 1. \tag{3}$$

Then, the distance between UAV and user is written as $s_{ij}(t) = \sqrt{\|q_j(t) - I_i\|^2}$. Here, the used channel loss model is the space loss model, and the signal propagation loss $h_{ij}(t)$ of user i to UAV j in time slot t is

$$h_{ij}(t) = \frac{\delta}{\left\|q_j(t) - I_i\right\|}, \tag{4}$$

where δ is channel power gain. The upload rate of user i at time slot j is

$$R_i^{(u)}(t) = \sum_{j=1}^{K} b_{ij}(t) \alpha_i^{(u)} \theta_i^{(u)} \mathrm{Blog}_2\left(1 + \frac{P_i^{(u)}(t) h_{ij}(t)}{N_0}\right), \tag{5}$$

where $p_i^{(u)}(t)$ represents user transmitting power, and $h_{ij}(t)$ is the signal propagation loss from user i to the UAV j, and N_0 is spatial noise. Similar to the uplink, the download rate $R_i^{(d)}(t)$ of user i is

$$R_i^{(d)}(t) = \sum_{j=1}^{K} b_{ij}(t) \alpha_i^{(d)} \theta_i^{(d)} \mathrm{Blog}_2\left(1 + \frac{P_{ij}^{(d)}(t) h_{ij}(t)}{N_0}\right), \tag{6}$$

where $p_{ij}^{(d)}(t)$ represents the transmit power allocated by the UAV j. Therefore, uploading data amount $D_i^{(u)}(t)$ and downloading data amount $D_i^{(d)}(t)$ for user i is written as

$$D_i^{(u)}(t) = \frac{\mathrm{TR}_i^{(u)}(t)}{M}, \qquad D_i^{(d)}(t) = \frac{\mathrm{TR}_i^{(d)}(t)}{M}, \tag{7}$$

$$D_i^{(\min)}(t) \le D_i^{(d)}(t). \tag{8}$$

Then, flight speed for UAV j in time slot t is given by

$$v_j(t) = \frac{\left\|\mathbf{q}_j(t) - \mathbf{q}_j(t-1)\right\|}{T/M}. \tag{9}$$

Moreover, due to the limitations of volume and power, flight speed of UAV is upper bounded by its maximum flight speed $v_j^{(\max)}$

$$v_j(t) \le v_j^{(\max)}. \tag{10}$$

Due to user equipment size and security factors, the user transmitting power has a certain upper bound

$$p_i^{(u)}(t) \le p_i^{(\max)}, \tag{11}$$

which is given by

$$\sum_{i=1}^{N} p_{ij}^{(d)}(t) \le p_j^{(\max)}, \tag{12}$$

for each use. To ensure transmission, both of the user and UAV transmitting powers should be greater than 0. Then, one has

$$0 < p_i^{(u)}, \tag{13}$$

$$0 < p_{ij}^{(d)}. \tag{14}$$

Since the UAV computing power is high [16], the calculation time and download time for UAVs are ignored. The energy consumed by UAVs includes flight energy consumption and communication energy consumption. The flight energy consumption $E_j^{(fly)}(t)$ of the UAV j is

$$E_j^{(fly)}(t) = \frac{0.5gT\left\|v_j(t)\right\|^2}{M}, \tag{15}$$

where g represents weight of UAV. At each time slot t, the calculation energy consumption model for UAV is created as

$$E_{ij}^{(c)}(t) = \varphi D_i^{(u)}(t) \gamma_j \left(f_{ij}^{(c)}\right)^2 \forall i \in K, j \in N, \tag{16}$$

where φ denotes energy conversion efficiency of UAV processor. $\gamma_i c$ represents the number of CPU cycles required

for user to calculate each bit of data, and $f_{ij}^{(c)}$ is CPU frequency that the UAV j is assigned to the user i in the time slot t. The calculation energy consumption of all the users for UAV j is

$$E_j^{(c)}(t) = \sum_{i=1}^{N} E_{ij}^{(c)}(t). \tag{17}$$

In time slot t, the download energy consumption generated by UAV j is

$$E_{ij}^{(d)}(t) = \frac{T\theta_i^{(d)} b_{ij}(t) p_{ij}^{(d)}(t) \alpha_i^{(d)}(t)}{M}. \tag{18}$$

For UAV j, the communication energy consumption caused by data download is

$$E_j^{(d)}(t) = \sum_{i=1}^{N} E_{ij}^{(d)}(t). \tag{19}$$

Similarly, in time slot t, the upload energy consumption generated by UAV j is

$$E_j^{(u)}(t) = \sum_{i=1}^{N} \frac{b_{ij} T\theta_i^{(u)} p_j^{(r)} \alpha_i^{(u)}(t)}{M}. \tag{20}$$

Due to effects such as UAV batteries and volume limitations, the energy for UAVs is limited, which cannot exceed the maximum residual energy σ_j in the UAV, and hence, the UAV has following energy constraints

$$\sum_{t=1}^{M} \left(E_j^{(fly)}(t) + E_j^{(d)}(t) + E_j^{(u)}(t) + E_j^{(c)}(t) \right) \le \sigma_j. \tag{21}$$

In addition, since multiple UAVs fly in the same height, collision avoiding is a problem that must be solved. $d_{\min}$ is defined as the safest distance between multiple UAVs. During each time slot, UAV i and UAV j must follow the conditions

$$\left\| \mathbf{q}_i(t) - \mathbf{q}_j(t) \right\| \ge d_{\min}. \tag{22}$$

Considering the fairness of users, the goal is to maximize minimum user computational rate. Let η represent the minimum calculation rate of all users. Then, we have

$$\sum_{t=1}^{M} D_i^{(u)}(t) \ge \eta. \tag{23}$$

Since the UAV has a fixed starting point $\mathbf{q}_j^{(s)}$ and an ending point $\mathbf{q}_j^{(e)}$, it has constraint

$$\mathbf{q}_j(0) = \mathbf{q}_j^{(s)}, \tag{24}$$

$$\mathbf{q}_j(M) = \mathbf{q}_j^{(e)}. \tag{25}$$

Therefore, the optimization P0 is formulated as

$$P0' \quad \max_{\alpha_i^{(u)}(t), \alpha_i^{(d)}(t), \mathbf{q}_j(t), p_i^{(u)}(t), p_{ij}^{(d)}(t), b_{ij}(t)} \eta s.t. b_{ij}(t) \in \{0, 1\}, \tag{26}$$

and equations (1)–(3), (9), (11)–(15) and (22)–(26)

The constraints (1) and (2) represent dynamic bandwidth allocation constraints while the constraint (3) represents the dynamic matching constraint for UAVs and the users. The constraint (8) is the minimum user download rate constraint, and the constraint (10) represents the maximum UAV flight rate constraint. The constraints (11) and (13) represent the upper and lower limits of transmission power of the user, respectively. The constraints (12) and (14) represent the transmission power constraints of the UAVs. (21) indicates the energy constraint of the UAV, while (22) represents multiple UAV anti-collision constraints, and (23) is minimum user upload rate constraint. In addition, the constraints (24) and (25) denotes a fixed starting point and ending point for the UAV. In order to ensure fairness between users, the objective function is taken to maximize minimum user calculation rate.

3. Three-Stage Iterative Optimization Algorithm

Since $b_{ij}(t)$ is a binary variable, $p_{ij}^{(d)}(t)$ and $\alpha_i^{(d)}(t)$ are continuous variables. Additionally, there is a nonlinear coupling between the variables in constraint (24). Thus, the problem $\mathbf{P}0'$ can be considered as a mixed integer nonlinear programming problem. At the same time, as the constraints (9) and (24) are non-convex, the problem $\mathbf{P}0'$ changes to be a non-convex optimization problem. For non-convex mixed integer nonlinear programming problems (MINLP), currently, there is no effective solution. Thus, it is difficult to solve this problem since multiple variables are coupled together. Herein, we propose a TSIO to solve problem $\mathbf{P}0'$. The first stage fixes $\mathbf{q}_j(t)$ and $b_{ij}(t)$. Then, this problem is transformed into a resource allocation optimization problem. The second stage takes the value of the resource allocation-related variable into $\mathbf{P}0'$. At the same time, $b_{ij}(t)$ are always fixed, and an optimization problem of UAV path planning and collision avoidance with only the variable $\mathbf{q}_j(t)$ can be obtained. The third stage brings $\mathbf{q}_j(t)$ and the resource allocation-related variables into $\mathbf{P}0'$. Then, the optimization problem has only integer variables.

3.1. Stage 1: Resource Optimization. The first step of the TSIO method is to fix the variables $\mathbf{q}_j(t), b_{ij}(t)$, and the following optimization problem $\mathbf{P}1'$ is obtained for UAV and user matching

$$\mathbf{P1}': \max_{\alpha_i^{(u)}(t), \alpha_i^{(d)}(t), p_i^{(u)}(t), p_{ij}^{(d)}(t)} \eta, \tag{27}$$

s.t. (1), (2), (9), (12)–(15) and (12)–(26).

The nonlinear coupling of variables exists in the constraints in (8) and (21) so that the problem P0 is a non-convex nonlinear programming problem. First, let $\phi_i^{(u)}(t) = \alpha_i^{(u)}(t) p_i^{(u)}(t)$, $\phi_{ij}^{(d)}(t) = \alpha_i^{(d)}(u) p_{ij}^{(d)}$, $\phi_{ij}^{(r)}(u) = \alpha_i^{(u)}(u) p_j^{(r)}$. The problem $\mathbf{P}1'$ can be changed to the following optimization problem $\mathbf{P}2'$ given in (28),

$$\mathbf{P2}': \max_{\alpha_i^{(u)}(t),\alpha_i^{(d)}(t),p_i^{(u)}(t),p_{ij}^{(d)}(t),\phi_{ij}^{(d)}(t),\phi_i^{(u)}(t),\phi_{ij}^{(r)}(t)} \eta s.t.\ (1),\ (2),\ (12)-(15)$$

$$\begin{aligned}
&\frac{T}{M}\sum_{j=1}^{K}\left(b_{ij}(t)\alpha_i^{(d)}(t)\theta_i^{(d)}B\log_2\left(1+\frac{\phi_{ij}^{(d)}(t)h_{ij}(t)}{\alpha_i^{(d)}(t)N_0}\right)\right)\geq D_i^{(\min)}(t)\\
&\frac{T}{M}\sum_{j=1}^{K}\left(b_{ij}(t)\alpha_i^{(u)}(t)\theta_i^{(u)}B\log_2\left(1+\frac{\phi_i^{(u)}(t)h_{ij}(t)}{\alpha_i^{(d)}(t)N_0}\right)\right)\geq \eta\\
&\sum_{t=1}^{M}\left(\frac{0.5T}{M}v_j^2(t)+\left(\sum_{i=1}^{N}\frac{T}{M}\theta_i^{(d)}b_{ij}(t)\phi_{ij}^{(d)}(t)\right)+\left(\sum_{i=1}^{N}\frac{T}{M}\theta_i^{(u)}b_{ij}(t)\phi_{ij}^{(r)}(t)\right)\right.\\
&\left.+\sum_{i=1}^{N}\frac{T\varphi\gamma_j\left(f_{ij}^{(c)}\right)^2}{M}\left(\sum_{j=1}^{K}\left(b_{ij}(t)\alpha_i^{(u)}(t)\theta_i^{(u)}B\log_2\left(1+\frac{\phi_i^{(u)}(t)h_{ij}(t)}{N_0\alpha_i^{(u)}(t)}\right)\right)\right)\right)\leq\sigma_j,
\end{aligned} \tag{28}$$

$$\begin{aligned}
L=\eta&+\sum_{t=1}^{M}\sum_{i=1}^{N}\left(\beta_i^{(u)}(t)\left(\sum_{i=1}^{N}\alpha_i^{(u)}(t)\theta_i^{(u)}-1\right)\right)+\sum_{t=1}^{M}\sum_{i=1}^{N}\left(\beta_i^{(d)}(t)\left(\sum_{i=1}^{N}\alpha_i^{(d)}(t)\theta_i^{(d)}-1\right)\right)\\
&+\sum_{i=1}^{N}\sum_{t=1}^{M}\pi_i^{(d)}(t)\left(D_i^{(\min)}(t)-\sum_{j=1}^{K}\xi_{ij}^{(d)}\right)+\sum_{t=1}^{M}\sum_{i=1}^{N}\left(\mu_i^{(um)}(t)\left(p_i^{(u)}(t)-p_i^{(\max)}\right)\right)\\
&+\sum_{t=1}^{M}\sum_{j=1}^{K}\left(\mu_j^{(dm)}(t)\left(\sum_{i=1}^{N}p_{ij}^{(d)}(t)-p_j^{(\max)}\right)\right)-\sum_{t=1}^{M}\sum_{i=1}^{N}\mu_i^{(d)}(t)p_i^{(d)}(t)-\sum_{t=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{K}\mu_{ij}^{(d)}(t)p_{ij}^{(d)}(t)+\sum_{i=1}^{N}\pi_i^{(u)}\left(\eta-\sum_{t=1}^{M}D_i^{(u)}(t)\right)\\
&+\sum_{j=1}^{K}\left(\lambda_j\left(\sum_{t=1}^{M}\left(E_j^{(fly)}(t)+\sum_{i=1}^{N}\left(\frac{T}{M}\theta_i^{(d)}b_{ij}(t)\phi_{ij}^{(d)}(t)\right)+\sum_{i=1}^{N}\left(\frac{T}{M}\theta_i^{(u)}b_{ij}(t)\phi_{ij}^{(r)}(t)\right)+\sum_{i=1}^{N}\frac{\varphi\gamma_j\left(f_{ij}^{(c)}\right)^2T}{M}\sum_{m=1}^{K}b_{im}(t)\xi_i^{(u)}(t)\right)\right)\right),
\end{aligned} \tag{29}$$

where these constraints have nonlinear coupling variables, resulting in the problem being a non-convex nonlinear programming problem. Let $\varphi_i^u(t)=\alpha_i^u(t)p_i^u(t)$, $\varphi_{ij}^d(t)=\alpha_i^d(t)p_{ij}^d(t)$, $\varphi_{ij}^{(r)}(t)=\alpha_i^{(u)}(t)p_j^{(r)}$. Then problem P1 can be transformed into the optimization problem P2. Since the constraints (1), (2), (12)–(15) and (31) and the objective function in the problem $\mathbf{P2}'$ are both linear functions, and the constraints (28) and (29) are both nonlinear convex, the problem $\mathbf{P2}'$ is considered as a convex optimization problem. It can be derived using convex optimization tools. Then, we get the following Theorem 1.

$$\kappa\left(\beta_i^{(u)}(t),\beta_i^{(d)}(t),\pi_i^{(d)}(t),\mu_i^{(um)}(t),\mu_j^{(dm)}(t),\mu_i^{(d)}(t),\mu_{ij}^{(d)}(t),\pi_i^{(u)},\lambda_j\right)=\max L, \tag{30}$$

$$\min\kappa\left(\beta_i^{(u)}(t),\beta_i^{(d)}(t),\pi_i^{(d)}(t),\mu_i^{(um)}(t),\mu_j^{(dm)}(t),\mu_i^{(d)}(t),\mu_{ij}^{(d)}(t),\pi_i^{(u)},\lambda_j\right). \tag{31}$$

Theorem 1. *In question* $\mathbf{P2}'$*, the expression of the optimal solution* $\alpha_{i,opt}^{(u)}(t),\alpha_{i,opt}^{(d)}(t),p_{i,opt}^{(u)}(t),p_{ij,opt}^{(d)}(t)$ *of the variable* $\alpha_i^{(u)}(t),\alpha_i^{(d)}(t),p_i^{(u)}(t),p_{ij}^{(d)}(t)$ *can be obtained by solving the Lagrangian multipliers corresponding to (1), (2), (12)–(15) and (29)–(31).*

Proof. To simplify the expression, let $\xi_{ij}^{(d)}(t)=T/M\alpha_i^{(d)}(t)\theta_i^{(d)}B\log_2(1+\phi_{ij}^{(d)}(t)h_{ij}(t)/\alpha_i^{(d)}(t)N_0)$, $\xi_i^{(u)}(t)=T/M\alpha_i^{(u)}(t)\theta_i^{(u)}B\log_2(1+\phi_i^{(u)}(t)h_i(t)/\alpha_i^{(u)}(t)N_0)$. Then, the Lagrangian function of the problem $\mathbf{P2}'$ can be constructed and given by equation (30), where $\beta_i^{(u)}(t),\beta_i^{(d)}(t),\pi_i^{(d)}(t),\mu_i^{(um)}(t),\mu_j^{(dm)}(t),\mu_i^{(d)}(t),\mu_{ij}^{(d)}(t),\pi_i^{(u)},\lambda_j$ are the corresponding Lagrangian multipliers for constraints (1), (2), (12)–(15) and (29)–(31), respectively. Next, Lagrangian dual function for $\mathbf{P2}'$ is presented as equation (31).

By solving its dual problem, the optimal solution of P2 can be obtained. The dual problem is given in (31) since $\mathbf{P2}'$ is a convex optimization problem. Then, the optimal solution can be obtained. □

3.2. Stage 2: Decision Optimization. When the solution of Stage 1 is completed, the optimal solution of $\alpha_{i,opt}^{(u)}(t), \alpha_{i,opt}^{(d)}(t), p_{i,opt}^{(u)}(t), p_{ij,opt}^{(d)}(t)$ is assigned to the variables $\alpha_i^{(u)}(t), \alpha_i^{(d)}(t), p_i^{(u)}(t), p_{ij}^{(d)}(t)$ and brought into the original question $\mathbf{P1}'$. Then, by fixing the variable $b_{ij}(t)$, we get the following multi-UAV path planning problem $\mathbf{P3}$

$$P3: \max_{\mathbf{q}_j(t)} \eta, \tag{32}$$

s.t (9), (11) and (24)–(26)

$$\frac{T}{M}\sum_{j=1}^{K}\left(b_{ij}(t)\alpha_i^{(d)}(t)\theta_i^{(d)}B\log_2\left(1+\frac{p_{ij}^{(d)}(t)\delta}{N_0\left(H^2\right)+\left\|q_j(t)-l_i(t)\right\|^2}\right)\right)\geq D_i^{(\min)}(t), \tag{33}$$

$$\frac{T}{M}\sum_{j=1}^{K}\left(b_{ij}(t)\alpha_i^{(d)}(t)\theta_i^{(d)}B\log_2\left(1+\frac{p_{ij}^{(d)}(t)\delta}{N_0\left(H^2\right)+\left\|q_j(t)-l_i(t)\right\|^2}\right)\right)\geq \eta, \tag{34}$$

$$\begin{aligned}\sum_{t=1}^{M}\Bigg(&0.5g\left(\left\|\mathbf{q}_j(t)-\mathbf{q}_j(t-1)\right\|\right)+E_j^{(d)}(t)+E_j^{(u)}(t)\\&+\sum_{i=1}^{N}\frac{T\varphi\gamma_j\left(f_{ij}^{(c)}\right)^2}{M}\left(\sum_{j=1}^{K}b_{ij}(t)\alpha_i^{(u)}(t)\theta_i^{(u)}B\times\log_2\left(1+\frac{p_i^{(u)}(t)\delta}{N_0\left(H^2+\left\|\mathbf{q}_j(t)-\mathbf{l}_i(t)\right\|^2\right)}\right)\right)\Bigg)\leq\sigma_j,\end{aligned} \tag{35}$$

where the constraints (9), (22), and (24) are listed in equations (34), (35), and (36). By finding the Hessian matrix of the function, the constraints (27)–(29) can be found to be convex. Therefore, the following Theorem 2 can be obtained

Theorem 2. *For any given feasible UAV trajectory* $\mathbf{q}_j^{(0)}(t)$*, the following inequality holds*

$$\begin{aligned}\log_2\left(1+\frac{\delta p_i^{(u)}(t)}{N_0\left(H^2+\left\|\mathbf{q}_j(t)-\mathbf{l}_i(t)\right\|^2\right)}\right)&\geq\psi_{ij}^{(u)}(t)=\log_2\left(1+\frac{\delta p_i^{(u)}(t)}{N_0\left(H^2+\left\|\mathbf{q}_j^{(0)}(t)-\mathbf{l}_i(t)\right\|^2\right)}\right)\\&-\left(\frac{\delta\log_2(e)p_i^{(u)}(t)\left(\left\|\mathbf{q}_j(t)-\mathbf{l}_i(t)\right\|^2\right)}{\left(N_0H^2+\delta p_{ij}^{(d)}(t)+N_0\left\|\mathbf{q}_j^{(0)}(t)\right\|^2\right)\left(H^2+\left\|\mathbf{q}_j^{(0)}(t)\right\|^2\right)}\right),\\\log_2\left(1+\frac{\delta p_{ij}^{(d)}(t)}{N_0\left(H^2+\left\|\mathbf{q}_j(t)-\mathbf{l}_i(t)\right\|^2\right)}\right)&\geq\psi_{ij}^{(d)}(t)=\log_2\left(1+\frac{\delta p_{ij}^{(d)}(t)}{N_0\left(H^2+\left\|\mathbf{q}_j^{(0)}(t)-\mathbf{l}_i(t)\right\|^2\right)}\right)\\&-\left(\frac{\delta\log_2(e)p_{ij}^{(d)}(t)\left(\left\|\mathbf{q}_j(t)-\mathbf{l}_i(t)\right\|^2\right)}{\left(N_0H^2+\delta p_{ij}^{(d)}(t)+N_0\left\|\mathbf{q}_j^{(0)}(t)\right\|^2\right)\left(H^2+\left\|\mathbf{q}_j^{(0)}(t)\right\|^2\right)}\right).\end{aligned} \tag{36}$$

When $q_j(t)=q_j^{(0)}(t)$, the equal sign of inequality (30) and (31) holds. Obviously, the proof can be gotten using Taylor formula, which is to say that after the non-convex term is relaxed, it becomes a convex function $\mathbf{P3}'$. If it is brought into a problem, the convex optimization problem $\mathbf{P4}'$ can be obtained as follows

$$\mathbf{P4}'\colon \max_{\mathbf{q}_j(t)} \eta s.t.\ (11)\ \frac{T}{M}\sum_{j=1}^{K}\left(b_{ij}(t)\alpha_i^{(d)}(t)\theta_i^{(d)}B\psi_{ij}^{(d)}(t)\right) \geq D_i^{(\min)}(t)\ \frac{T}{M}\sum_{j=1}^{K}\left(b_{ij}(t)\alpha_i^{(u)}(t)\theta_i^{(u)}B\psi_{ij}^{(u)}(t)\right) \geq \eta$$

$$\sum_{t=1}^{M}\left(0.5g\left(\left\|\mathbf{q}_j(t)-\mathbf{q}_j(t-1)\right\|\right)+E_j^{(d)}(t)+E_j^{(u)}(t)+\sum_{i=1}^{N}\frac{T\varphi\gamma_j\left(f_{ij}^{(c)}\right)^2}{M}\left(\sum_{j=1}^{K}\left(b_{ij}(t)\alpha_i^{(u)}(t)\theta_i^{(u)}B\psi_{ij}^{(u)}(t)\right)\right)\right)\leq \sigma_j. \tag{37}$$

Since all constraints and objective functions in $\mathbf{P4}'$ are convex, $\mathbf{P4}'$ is a convex optimization problem. For this problem, we can solve it using convex optimization tool.

3.3. Stage 3: Trajectory Optimization. When the optimal solution of Stage 1 and Stage 2 are solved, the following optimization problem **P**5 of UAV for user matching can be obtained

$$\log_2\left(1+\frac{\delta p_{ij}^{(d)}(t)}{N_0\left(H^2+\left\|\mathbf{q}_j(t)-\mathbf{l}_i(t)\right\|^2\right)}\right)\geq \psi_{ij}^{(d)}(t)=\log_2\left(1+\frac{\delta p_{ij}^{(d)}(t)}{N_0\left(H^2+\left\|\mathbf{q}_j^{(0)}(t)-\mathbf{l}_i(t)\right\|^2\right)}\right)$$
$$-\left(\frac{\delta\log_2(e)p_{ij}^{(d)}(t)\left(\left\|\mathbf{q}_j(t)-\mathbf{l}_i(t)\right\|^2\right)}{\left(N_0H^2+\delta p_{ij}^{(d)}(t)+N_0\left\|\mathbf{q}_j^{(0)}(t)\right\|^2\right)\left(H^2+\left\|\mathbf{q}_j^{(0)}(t)\right\|^2\right)}\right). \tag{38}$$

When $q_j(t)=q_j^{(0)}(t)$, sign of equality in (29) and (30) holds. Obviously, the proof can be obtained using Taylor formula, which is to say that it becomes a convex function $\mathbf{P3}'$ after non-convex term relaxation. Then, the convex optimization problem **P**5 can be obtained

$$P5\colon \max_{b_{ij}(t)} \eta, \tag{39}$$

s.t (1), (2), (9), (22) and (24).

Where $b_{ij}(t)$ is a binary variable, and both constraint and objective function are linear so that the problem **P**5 is an integer linear programming problem (ILP) which can be solved using the classical branch and bound (BB) algorithm [17]. The main idea of the BB algorithm is to continuously traverse the solution space of **P**5 until optimal solution is found. The integer variable is relaxed to get a continuous variable to create a new sub-problem with a branch operation. Meanwhile, the optimal solution based on the sub-problem continuously obtains upper and lower bounds. When the upper and lower bounds are equal, the optimal solution is gotten. When the optimal solution $b_{ij,opt}(t)$ of the TSIO and η_{opt} of the objective function are obtained. They will be brought into the first stage to promote to update the phases until the difference between optimal values is less than a threshold to stop the iteration. In a word, the solution is converged and an approximate optimal solution is found. The devised TSIO algorithm is summarized as Algorithm 1.

4. Results and Discussion

In this section, the performance of the proposed multi-UAV-assisted mobile offloading using devised TSIO algorithm is presented, analyzed and discussed in detail. Herein, assuming that the number of users is $N=5$ and the coordinates for all users are $L_1=[0,0]$, $L_2=[10,0]$, $L_3=[0,10]$, $L_4=[10,10]$, and $L_5=[5,5]$, respectively. Also, all users have the computation requirement $\mathbf{a}^{(u)}=[1,1,1,1,1]$ and download requirements $\mathbf{a}^{(d)}=[1,1,1,1,1]$, where the value equals to 1, meaning that the user has the requirement. Two UAVs are used, namely, UAV1 and UAV2, respectively. The start point of UAV1 is the same as L1, and the target point (destination) is the same as L2, while the start point of UAV2 is the same as L4, and the target point (destination) is the same as L3. UAV's maximum flight speed is 20m/s. The continuous flight time $T=2s$, which is divided into 50 time slots. The vertical height of the UAVs is $H=15$m. The power of UAV1 is $500kJ^2$ and the power of UAV2 is $400kJ^2$, where the power of UAV1 is greater than that of UAV2. Experimental UAV flight trajectory and user matching are given using computer simulations, and the effects of different optimization schemes on the minimum computing rate are studied and discussed, which verifies the superior performance of the proposed scheme and TSIO algorithm.

Figure 2 shows the flight trajectory for UAV1 and UAV2, while Table 1 presents the UAV used by users. From Table 1, we can see that UAV1 provides service for L1, L2, and L5, while UAV2 provides service for L3 and L4. The trajectory of UAV1 is approximately elliptical, but UAV2 flies along a straight line. The main reason for different trajectories is that UAV1 serves to three users to ensure that all three users could get a reasonable transmission rate. In order to ensure the computation and download rate at L5, UAV1 needs to approach to L5 initially to reduce transmission distance and increase transmission rate. The larger vertical height between UAV1 and L5 is, the larger distances between UAV1, L1, and L2 are. Therefore, when the vertical distance of UAV1 rises to a certain height, it will not change so that it

(1) Initialize $\mathbf{q}_j^{(s)}(t)$, $\mathbf{q}_j^{(e)}(t)$, $k = 1$, threshold ρ_1, ρ_2, ρ_3, $\eta^0 = \infty$
(2) while $\eta^k - \eta^{k-1} > \rho_1$ do
(3) Given $\mathbf{q}_{j,opt}$ and variable $b_{ij,opt}$, use convex optimization tool to solve P2, and obtain $\alpha_{i,opt}^{(u)}(t), \alpha_{i,opt}^{(d)}(t), p_{i,opt}^{(u)}(t), p_{ij,opt}^{(d)}(t)$;
(4) Substitute the optimal solution of P2 into P4, and let $m = 1$;
(5) while $\exists j \in \{1, 2, \ldots, K\}, \sum_{t=1}^{M} \|\mathbf{q}_{j,m}(t) - \mathbf{q}_{j,m-1}(t)\| > \rho_2$ do
(6) Solve $P4$ using convex optimization tool to obtain $\mathbf{q}_{j,opt}$; $m = m + 1$; $\mathbf{q}_{j,m} = \mathbf{q}_{j,opt}$
(7) end while
(8) Substitute $\mathbf{q}_{j,opt}$ of $P4$ into $P5$; relax b_{ij} of P5 to grt a continuous variable. Then, obtain its relaxation sub-problem SP_k and place them in the solution queue a; Set upper bound of
(9) while list is not empty do
(10) Take a sub-problem from the solution queue SP_k, solve it, and get its optimal solution $b_{ij,opt}$ and η_{opt};
(11) if b_{ij}^m are integers then
(12) if $\eta_{opt} > LB$ then
(13) Set $LB = \eta_{opt}$, remove the sub-problems if $UB < LB$
(14) end if
(15) else
(16) Relax $b_{ij,opt}$, then, get a new sub-problem SP_{k+1}, SP_{k+2} and put it into the solution queue; Set $m = m + 1, UB^m = \eta_{opt}$
(17) end if
(18) end while
(19) $k = k + 1, \eta^k = \eta_{opt}$
(20) end while

Algorithm 1: The TSIO algorithm.

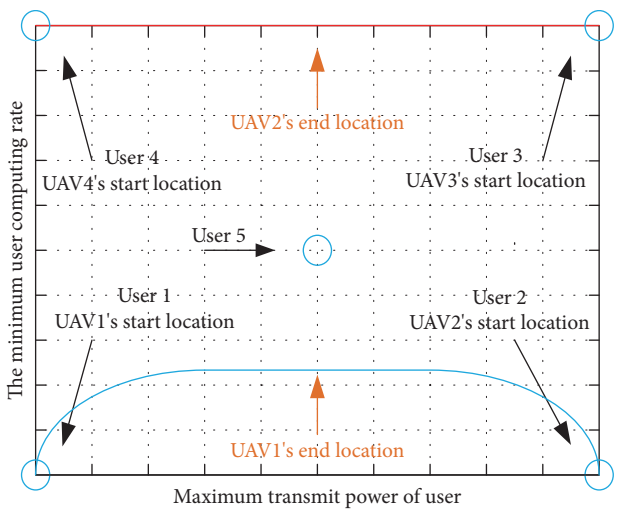


Figure 2: UAV1 and UAV2 flight trajectory.

Table 1: User matches with drone.

	user1	user2	user3	user4	user5
UAV num	1	1	2	2	1

can guarantee the transmission rate for L1 and L2. Thus, the flight trajectory of UAV1 is affected by three users, namely, L1, L2, and L5. On the contrary, UAV 2 is moved horizontally between L3, L4 to get the shortest distance from L3 and L4 since UAV2 provides services for L3 and L4 to increase transmission rate and save transmission power. L1 and L2 choose UAV1 for service because they are close to each other. Similar to the L3 and L4, L5 chooses UAV1 for service as the power of UAV1 is larger than that of UAV2. Hence, the trajectory of UAV and the choice of users are related to not only the distance but also the UAV power.

Figure 3 reveals the dynamic change of the user's minimum computing rate in the process of solving the TSIO algorithm. It is found that the initial minimum computing rate of users is relatively high, and then it decreases until convergence. Herein, the TSIO algorithm is divided into three stages to solve three problems, which cannot satisfy all constraints at the initial state. With the algorithm going, all the constraints are gradually satisfied. When all constraints are satisfied simultaneously, the results do not change. It is observed that the convergence speed for the TSIO algorithm is faster, which converges at about 13 times. Also, the greater maximum user transmission power is, the greater minimum user calculation rate is and the higher signal-to-noise ratio (SNR) of the transmission is, which further increases the upload rate. Meanwhile, all the users can get high minimum calculation rate to ensure the fairness between users.

Figure 4 demonstrates the change of the minimum user calculation rate with maximum user transmit power in different UAV paths. It can be seen that the larger the maximum user transmit power is, the larger the minimum calculation rate is, which is same as the conclusion in Figure 3. Additionally, compared with the semi-circular path with fixed UAVs, the optimal path obtained by the proposed scheme can achieve a higher minimum calculation rate.

Figure 5 illustrates the variation of minimum user computation rate with maximum transmit power for UAV1 and UAV2 under different paths, where we assume that the maximum transmit powers for UAV1 and UAV2 are same. We found that the larger the maximum UAV transmitting power is, the larger the minimum user computing rate is. With the increasing of the maximum UAV transmit power,

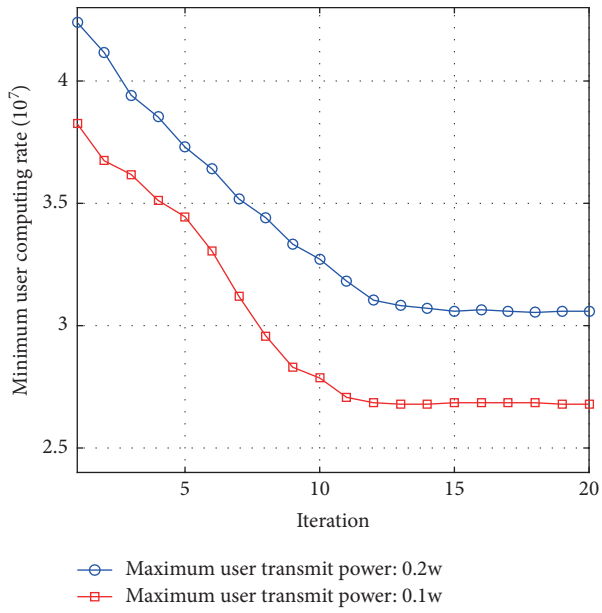


Figure 3: Dynamic change of minimum user calculation rate.

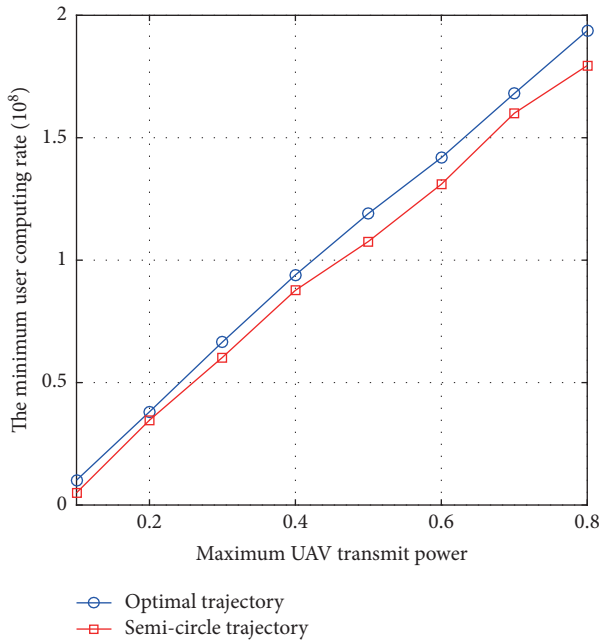


Figure 4: Relation between minimum user computing rate and maximum user transmitting Power.

the transmission power allocating to each user also increases, which further increases SNR. Therefore, UAV can satisfy minimum download rate constraining of users in a longer distance. Because UAV path is less affected by user's minimum download rate constraint, UAV path can be further optimized to make user upload rate higher. Hence, the higher the maximum UAV transmit power is, the higher the minimum user download rate is. Moreover, we can see that

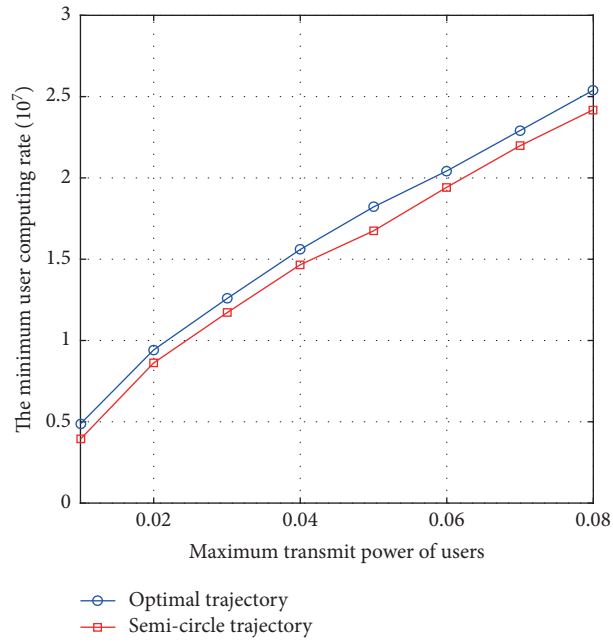


Figure 5: Relation between minimum UAV computing rate and maximum UAV transmitting Power.

the optimal path proposed in this paper can achieve better results than that of the semi-circular path.

Figure 6 shows the user transmit power dynamically changes at different times. It is found that the transmitting power of L1 increases with the increment of time slot since L1 chooses UAV1 for service and the distance between L1 and UAV1 is smaller at beginning. In this case, users can achieve higher upload rate with low transmission power. As the UAV1 moves to L2, the distance between them increases. To maintain a higher upload rate, transmit power for L1 should be increased. At the same time, it is observed that L2 has a higher transmission power at its initial. Similar to user 1, it requires a larger transmit power to maintain a higher upload rate since L2 is far away from UAV1. As UAV1 moves, the distance between them becomes smaller and smaller, and it can reach same upload rate with a smaller transmission power. L3 and L4 select UAV2 for service, and the trend is similar to L1 and L2. However, the transmission power of L3 and L4 is smaller than L1 and L2 because UAV1 needs to serve L5 via changing its flight trajectory so that the distance between UAV1 and L1 and L2 becomes farther during the flight. UAV2 only serves L3 and L4, and their distance is relatively closer. Therefore, it only needs to use smaller transmit power to achieve the same upload rate like L1 and L2. Finally, it can be seen that the transmission power of L5 decreases firstly and then increases, which is related to the distance between users and UAV. From the UAV trajectory diagram, we can see that the distance between L5 and UAV1 is first decreased and then increased.

Figure 7 shows the dynamic variation of user transmission time ratio over different time slots. It can be seen that the proportion of transmission time for user 1 is continuously decreasing, which is similar to that of Figure 6. Since the distance between UAV1 and user decreases as the time slot

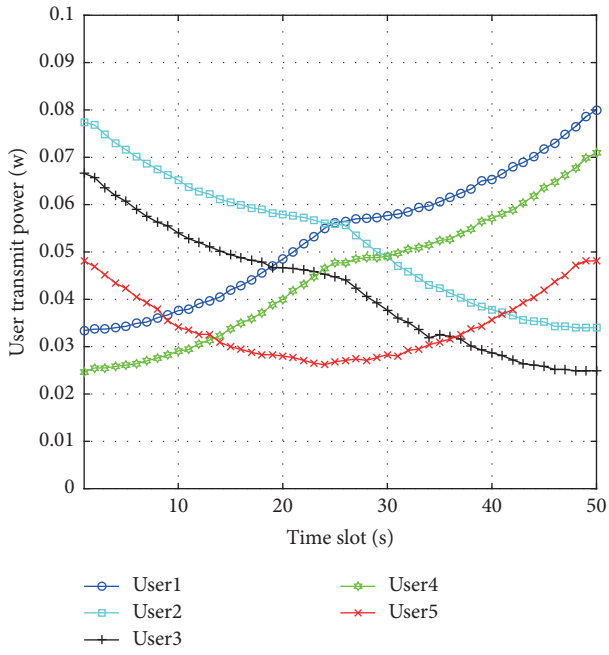


Figure 6: Dynamic changes of user transmit power in different time slots.

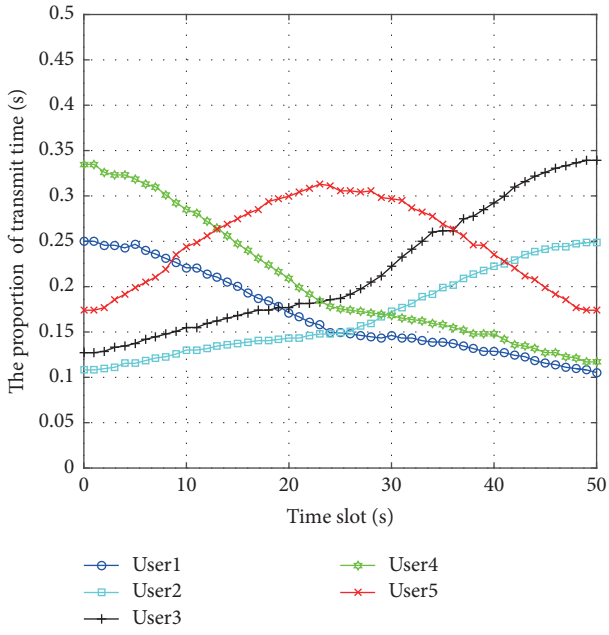


Figure 7: Dynamic change of user transmission time ratio in different slots.

increases, the user transmit power also increases. When the user transmit power is small, it indicates that the user is closer to UAV. In order to ensure the fairness between users, the UAV allocates less transmission time for more recent users and allocates more transmission time for farther users to ensure that all users can achieve minimum upload rate. Comparing Figure 6 with 7, it can be concluded that the farther the UAV is from the user, the greater the user transmit power and proportion of transmission time allocated by the

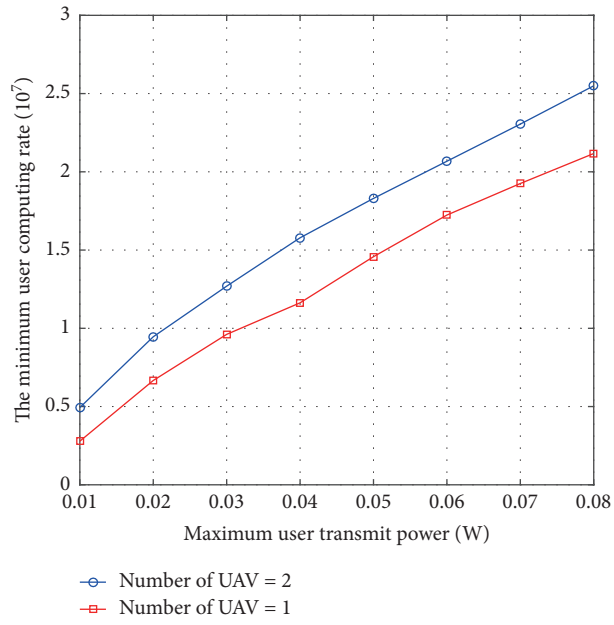


Figure 8: Dynamic changes of minimum computing rate with maximum transmit power for different UAVs.

user are. The reason for the change trend of the remaining users is similar to that of User 1, which is not described here.

Figure 8 presents the variation of the minimum user computational rate with maximum user transmit power for different UAV numbers. Firstly, similar to Figure 3, the greater the user transmit power is, the greater the user minimum computation rate is. Secondly, it can be seen that the more UAVs are, the greater the minimum user calculation rate is. As the number of UAVs increases, the competition between users becomes smaller, and the number of users of a single UAV service becomes smaller. Thus, the movement trajectory can be further optimized, and if the user is closer to UAV during the movement, the user can obtain a larger calculation rate with the same transmission power. Therefore, the greater the number of UAVs is, the greater the minimum user computational rate is, which also indicates the effectiveness of the proposed scheme.

5. Conclusion

A multi-UAV-assisting data unloading and computational uploading scheme for multi-user has been proposed, analyzed, and discussed, which is modeled as mixed-integer nonlinear optimization (MINO) problem. Also, a three-stage iterative optimization (TSIO) algorithm is presented, investigated and discussed to find a solution for the MINO problem. The path conflict avoidance between multiple UAVs, the matching between multiple UAV and users, and the effects, such as UAV energy, user uplink and downlink transmission bandwidth allocation have been investigated using simulation experiments. Additionally, the TSIO is also used for achieving maximization the minimum user calculation rate. Experimental results further verified the effectiveness of the proposed multi-UAV scheme to assist users for mobile offloading, which

also show that our proposed algorithm and solution is superior to the single UAV method.

We believe that with the proposed method, we solved the computing offloading and data offloading of fixed users by using UAV. However, in certain specific scenarios, the user is mobile (for example, search and rescue in disaster events). For this reason, we will solve the UAV cooperative communication and calculation under the condition of user movement in the follow-up work.

Data Availability

Basic data can be obtained from the corresponding author when needed.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Science and Technology Major Project of China (2016ZX03001023-005) and Fundamental Research Funds for the Central Universities (3072020CF0603).

References

[1] H. Z. Guo and J. J. Liu, "UAV-enhanced intelligent offloading for Internet of things at the edge," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2737–2746, 2020.

[2] J. Zhang, Z. Zhou, F. H. Zhou, and B. C. Seet, H. Zhang, Z. Cai, J. Wei, Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2114–2125, 2020.

[3] L. Zhao, K. Yang, Z. Tan, and S. Sharma, "A novel cost optimization strategy for SDN-enabled UAV-assisted vehicular computation offloading," *IIEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3664–3674, 2020.

[4] Y. Liu, K. Xiong, Q. Ni, and K. Ben Letaief, "UAV-assisted wireless powered cooperative mobile edge computing: joint offloading, CPU control, and trajectory optimization," *IEEE Internet of Things Journal*, vol. 22, no. 7, pp. 2777–2790, 2020.

[5] Z. J. Yu, Y. M. Gong, and S. M. Gong, Y. Guo, Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3147–3159, 2020.

[6] M. Hua, Y. Huang, Y. Wang, Q. Wu, H. Dai, L. Yang, Energy optimization for cellular-connected multi-UAV mobile edge computing systems with multi-access schemes," *Journal of Communications and Information Networks*, vol. 3, no. 4, pp. 33–44, 2018.

[7] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

[8] S. Chao, T. Chang, and J. Gong, "Multi-UAV interference coordination via joint trajectory and power control," *IEEE Transactions on Signal Processing*, vol. 68, pp. 843–858, 2020.

[9] C. Nan, W. C. Xu, and W. S. Shi, "Air-ground integrated mobile edge networks: architecture, challenges, and opportunities," *IEEE Communications Magazine*, vol. 56, pp. 26–32, 2018.

[10] M. Mozaffari and W. Saad, M. Bennis, Y.-H. Nam, M. Debbah, A tutorial on UAVs for wireless networks: applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[11] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 2049–2063, 2017.

[12] Y. Zhang and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[13] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[14] D. W. Matolak and R. Ruoyu Sun, "Unmanned aircraft systems: air-ground channel characterization for future applications," *IEEE Vehicular Technology Magazine*, vol. 10, no. 2, pp. 79–85, 2015.

[15] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of things services: comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.

[16] F. Wang, J. Xu, and X. Wang, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784–1797, 2020.

[17] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: a survey," *Operations Research*, vol. 14, no. 4, pp. 699–719, 1966.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4436824, 15 pages
https://doi.org/10.1155/2022/4436824

Research Article

A Practical Anonymous Voting Scheme Based on Blockchain for Internet of Energy

Houpeng Hu,[1] Jiaxiang Ou,[1] Bin Qian,[2] Yi Luo ,[2] Peilin He,[1] Mi Zhou,[2] and Zerui Chen[1]

[1]*Guizhou Power Grid Co. Ltd, Guiyang, China*
[2]*Institute of Metrology Technology Electric Power Research Institute CSG, Guangdong Provincial Key Laboratory of Intelligent Measurement and Advanced, Metering of Power Grid, Guangzhou, China*

Correspondence should be addressed to Yi Luo; luoyi_csg@outlook.com

Received 26 April 2022; Revised 15 June 2022; Accepted 5 July 2022; Published 21 August 2022

Academic Editor: Zheng Yang

E-voting allows us to build a democratic business in most Internet of things (IoT) systems. For example, we may vote to choose a proper energy broker in a smart grid system. In this study, we focus on e-voting services in an Internet of energy (IoE) system, which is a new-style smart grid. A practical e-voting in IoE may focus on the properties of fairness, decentralization, eligibility, anonymity, compatibility, verifiability, and coercion resistance. It is difficult to fulfil all these properties simultaneously. Traditional voting schemes often use a public bulletin board or administrator in the voting process, which makes them become centralized. Services that offer e-voting via blockchain can make the voting schemes decentralized. However, many of them ignore the complexity of organizing the data of the transactions, which should be confirmed by the miners. Moreover, to the best of the authors' knowledge, no works have tested the performance in the blockchain while considering practical use cases and constraints. Concerning all the challenges, we propose a practical anonymous voting scheme for IoE called IoEPAV. The proposed scheme fulfils all the mentioned design goals simultaneously. We tested IoEPAV both in different test networks of the Ethereum blockchain to give an overall evaluation. The practical evaluation can show that the proposed scheme is easy to be integrated into a real system like IoE. We also gave a comparison analysis with the state-of-the-art blockchain-based e-voting. All the results show that IoEPAV is decentralized, verifiable, anonymous, and highly efficient.

1. Introduction

The rapid development of the Internet of things (IoT)[1] promotes the development of various new intelligent network systems, i.e., smart power grid system [2]. Smart grid systems have led to a modern power network called the Internet of energy (IoE), which has drawn great interest from many countries [3]. As a kind of IoT, IoE is reshaping the energy industry into a smart industry with features of data-driven decision-making. However, the intrinsic features of IoE raise a number of challenges, such as autonomy, privacy, and decentralization [4–6]. One of the most common activities/applications in IoE is voting to make a decision.

In this study, we focus on the problem of designing a voting scheme fit for an IoE system. For instance, we may vote to choose a proper energy broker in IoE system. Recently, e-voting has become attractive [7] for its convenience in building a democratic activity/business. Voting schemes in an online way called e-voting have been studied by both academic world and industry world [7, 8]. Here, we formally state the design goals, which a deemed secure e-voting scheme must hold [9]. Moreover, we extend the design goals, which particularly are required in IoE [4]. All of them are as follows:

(i) Fairness. Fair voting should assure that no one can obtain the ballot results of others before he/she has submitted his/her ballot. It means the choice of a voter cannot be influenced by those who have voted ahead.

(ii) Decentralization. Any kind of trusted third party (TTP) such as election administrators and (independent) observers should be eliminated from the voting scheme.

(iii) Eligibility. The right of a voter should be checked before he/she begins to vote. To address this, most voting schemes will verify the identities of the voters at the beginning. Moreover, every voter can cast their vote only once.

(iv) Anonymity. The privacy of voters should be protected to make sure that no one can know the owner of a ballot from the voting result at the end.

(v) Compatibility. The voting scheme should be as simple as possible to be integrated into an IoE system.

(vi) Verifiability. Contrasted with the "Anonymity" property, verifiability guarantees that all the stages of the voting can be audited by the voters. For instance, a voter should be able to check whether his/her vote has been tallied or not. Moreover, the validity of each vote should be able to be verified by anyone. It seems to be a contradiction of the "Anonymity" property, and it is difficult for a voting scheme to acquire the two properties at the same time. We will show how to address this in our scheme later.

(vii) Coercion Resistance. To avoid anyone trying to coerce the voters to vote by following their instructions, a voting scheme should be coercion-resistant.

Fairness and eligibility are essential properties, which any voting scheme should fulfil. Besides, it is important to handle voting without any kind of trusted third party (TTP) in IoE due to its open and distributed features. This is a challenge for traditional e-voting schemes [10–12]. Most of them assumed that there are administrators or authorities implemented by a Web server to provide a consistent view of the results. As a result, a trusted third party is involved. Unfortunately, with a trusted third party, the protocols will be subjected to the single point of failure and are not available for a trustless environment.

Fortunately, blockchain technology offers a novel way to address the challenges of IoE [13]. There are already voting schemes built based on a blockchain network [14–16]. However, new challenges are raised to design a decentralized voting protocol via blockchain in IoE [13, 17, 18]. Firstly, it is difficult to provide verifiability along with anonymity, which seems two contradictory design goals for blockchain. Secondly, the voting scheme as a basic service in IoE should be compatible with the system. Some of the proposals assume that a voter can organize the data structure of a transaction unboundedly and it seems impossible for the existed blockchain networks. We say they are not practical. Thirdly, most works use theoretical analysis without real-world generalizable experiments, and to the best of our knowledge, no works have tested the performance while considering practical use cases and constraints.

Briefly, it is hard to find a solution that fulfils all the design goals mentioned above. To address this, our contributions are summarized as follows:

(i) We propose a blockchain-based decentralized anonymous voting scheme for the Internet of energy. To the best of our knowledge, our voting scheme called IoEPAV is the first work to take the key features of the IoE into account. Theoretical analysis is given to show that our proposed scheme fulfils the seven formally stated design goals and approaches to resist all the attacks in the threat models.

(ii) To make the proposed scheme practical, we use smart contracts to automate the voting process of the Internet of energy. With smart contracts, the voting scheme can be easy to integrate into the IoE system. A voter can follow the voting protocol by invoking the interfaces of the smart contracts. Any blockchain system including Ethereum 2.0, which supports smart contracts, is feasible for the proposed scheme, and we do not need to construct a whole new blockchain platform.

(iii) Compared with Yang's state-of-the-art blockchain-based scheme, our scheme enjoys both decentralization and fewer cryptographic operations; thereafter, we conduct experiments both in the development network and two live testnet of the Ethereum blockchain and the experiment shows that we have implemented a simple, effective, accurate, and low-cost decentralized trusted anonymous voting scheme.

The rest of this study is arranged as follows. In Section 2, we give the related work of the e-voting service. In Section 3, we introduce the necessary preliminary knowledge. In Section 4, we give our system model and security analysis. In Section 6, we provide an evaluation of the development network and testnet of the Ethereum blockchain. Finally, in Section 7 we draw a brief conclusion.

2. Related Work

To address the problem of large-scale elections, Fujioka et al. [12] presented a classic voting protocol in 1992. Their voting scheme is thought to be practical and solves the privacy and fairness problems. Thereafter, Ohkubo et al. [19] tried to decrease the voting round complexity to get a more convenient voting scheme for the voters than that presented by Fujioka. They introduced a kind of distributed talliers in their scheme. As an extension of the voting scheme proposed by Fujioka, a new coercion-resistant voting scheme offered by [20] in 2017 is provided and is an efficient scheme.

Since a kind of trusted third party (TTP) such as election administrators and (independent) observers was introduced in these practical voting schemes, solutions using blockchain technology have become a refreshed framework for voters to address the issues of fraud and corruption. In 2018, Srivastava et al. [21] proposed a voting model via blockchain to alleviate known problems in voting systems. Follow My Vote [14] provides a secure online voting platform based on blockchain. Follow My Vote has the capacity to audit the ballot box and watch the real-time voting progress. Another

organization Agora [15] proposed a digital voting system using blockchain, where votes will be recorded to various layers, assuring that the voting result could not have been tampered with. Braghin et al. [22] studied various consensus algorithms and cryptographic primitives such as homomorphic encryption and one-time ring signature, which solved the cryptographic problem of security conflicts, thus improving the security of voting system and making voting system more secure in a wider range.

To ensure security, privacy, and public verifiability of the whole progress, [16, 23] presented a new voting protocol that does not rely on any TTP. In [16], the authors employ a novel encryption mechanism to encrypt each vote. Proofs are generated for each encrypted vote as well. All the proofs will be stored in a blockchain, and everyone can check the validity of these proofs. They provided a performance and security analysis, which is claimed to show that the voting protocol is feasible for real-life elections. However, the implementation does not include the part interacting with the blockchain, and we cannot see the results in a real system. Table 1 gives an overall comparison of similar systems.

3. Preliminaries

3.1. Commitments and Voting. Commitment also called cryptographic commitment [24] is an important cryptographic primitive that has many applications. Here is an example to show the relationship between commitments and votes. Think about a situation that Voter 1 and Voter 2 decide to participate in a voting behavior along with others. In this scenario, the voting institution responsible for counting the votes uses a sealed vote, which generally works as follows: each voter submits a secret sealed vote for the candidate. Once all the votes have been cast, they can be counted. The voting mechanism has good game-theoretic properties, provided that voters do not collude and do not know each other's votes until the voting close. Therefore, a sealed vote is required.

Next, we consider how to do sealed voting when some voters are of outfield and communicate with the voting institution throughout the Internet. Here, a cryptographic commitment helps. To make it simple, we assume that there are two parties in a commitment scheme Commit. Let Alice be one party, who can firstly publish a string c as a commitment for a message $m \in M$. Then, with the property of cryptographic commitment, Alice can make other party, i.e., Bob, believe that the committed message was m, by opening the commitment. Generally speaking, a cryptographic commitment Commit consists of algorithms (C, V) that

(i) On input $m \in M$, the message to be committed, Algorithm $C(m)$ outputs two strings (c, o), and we call c the commitment string and o the opening string.

(ii) On input $m \in M$ and (c, o), Algorithm V outputs accept or reject indicating whether the committed message was m.

Alice firstly inputs a message $m \in m$ and calls Algorithm C to calculate (c, o). She sends the commitment string c to Bob and keeps the opening string o secret. Later, when Alice wants to open the commitment, she sends Bob m and o. Finally, Bob can verify whether the committed message was m by running Algorithm V.

A secure cryptographic commitment scheme Commit is required to satisfy the following two properties:

(i) Binding. Binding requires that a commitment does not disclose any additional information about the message. In particular, assume the adversary $\mathscr{A}$ outputs a 5-tuple (c, m_1, o_1, m_2, o_2), and we require the advantage that $(\text{BINDadv}[\mathscr{A}, C]: = \Pr[m_1 \neq m_2)$ and $V(m_1, c, o_1) = V(m_2, c, o_2) = \text{accept}]$ is negligible.

(ii) Hiding. Hiding requires that different messages do not produce the same commitment.

We use semantic security definition to formalize this. In particular, two games are performed between an adversary $\mathscr{A}$ and a challenger, denoted as Game 0 and Game 1. Let $b = 0, 1$, in the Game b, and the adversary $\mathscr{A}$ first outputs $m_0, m_1 \in M$, inputs a message $m_b \in M$, calls Algorithm C to calculate (c, o), and passes c to $\mathscr{A}$. Finally, $\mathscr{A}$ outputs a guess $\widehat{b} \in 0, 1$. For $b = 0, 1$, W_b is defined as the case, where $\mathscr{A}$ outputs 1 in Game b. We require that the advantage that

$$\text{BINDadv}[\mathscr{A}, C]: = \left|\Pr[W_0] - \Pr[W_1]\right|, \tag{1}$$

is negligible.

3.2. Blind Signature. According to the description in [25, 26], a cryptography blind signature is a kind of digital signature where the original message should be blinded (disguised) before being signed. Then, the signer will sign on the blinded message in a way like a conventional digital signature and output a blind signature. Then, the requester can generate a corresponding signature for the original message. In the end, the signature can be verified by everyone in a way like a conventional digital signature. The technique is usually used to provide a kind of privacy protection when the message requester and signer are not the same. For example, blind signatures can be used in an election system.

In a general signature scheme illustrated in Figure 1, the signer produces a digital signature on known message content. Compared with the general signature scheme, the process of blind signature [27] is as illustrated in Figure 2. The requester performs a blinding shift on the message before sending it to the signer. The signer who then signs on the blinded message will generate a blind signature and send it to the requester.

A general signature scheme is shown in Figure 1, in which a signer can generate a digital signature on a known message. Unlike a signature scheme, the process of blind signature is shown in Figure 2, in which the requester first blinds the message before it is sent to the signer, and then, the signer signs the blind message and sends a blind signature to the requester. With the blind signature, the requester can generate an unblinded signature for the original message.

Table 1: Property comparison.

Protocol	Fairness	Decentralization	Eligibility	Anonymity	Compatibility	Verifiability	Coercion resistance
Fujioka et al. [12]	✓		✓	✓		✓	
Ohkubo et al. [19]	✓		✓	✓		✓	
Grontas et al. [20]	✓		✓	✓		✓	✓
Follow My Vote [14]	✓	✓	✓			✓	
Yang et al. [16]	✓	✓	✓	✓		✓	✓
IoEPAV	✓	✓	✓	✓	✓	✓	✓

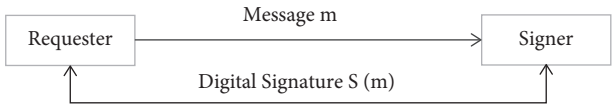


Figure 1: A general signature scheme.

In our voting scheme, we use the blind signature scheme of elliptic curve cryptography (ECC) and adopt the secp256k1 [28] elliptic curve. Elliptic curve domain parameters over $\mathbb{F}_p$ are a sextuple: $T(p, a, b, G, n, h)$ consisting of an integer p specifying the finite field $\mathbb{F}_p$ and two elements $a, b \in \mathbb{F}_p$ specifying an elliptic curve $E(\mathbb{F}_p)$ defined by the following equation:

$$E: y^2 \equiv x^3 + ax + b \pmod{p}. \tag{2}$$

G is a base point on $E(\mathbb{F}_p)$, n is the order of G, h is the cofactor where $h = \#E(\mathbb{F}_p)/n$, and $\mathbb{Z}_n$ represents the integer not more than n. Suppose that (d, P) is an asymmetric key pair of the signer, the message is m, and all else is as has been defined.

In particular, we consider the blind signature algorithm provided by Zhang et al. [27] as follows:

(i) (1) Let $k \in \mathbb{Z}_n$ be an integer randomly selected by the signer, which calculates $R = kG$. Then, the signer sends R to the requester.

(ii) (2) Firstly, the requester selects two integers γ and $\delta \in \mathbb{Z}_n$ randomly and computes $A = kG + \gamma G + \delta P = (x, y), t = x \bmod n$. It checks whether t equals zero. If so, the requester reselects γ and δ. Then, it computes $c = SHA256(m\|t)$ and $\hat{c} = c - \delta$; here, SHA256 [29] is a cryptography hash function. Finally, the requester sends $\hat{c}$ to the signer as the blinded message.

(iii) (3) The signer generates a blind signature $\hat{s} = k - \hat{c}d$ using the blinded message and sends to the requester.

(iv) (4) On receiving $\hat{s}$, the requester computes $s = \hat{s} + \gamma$, and along with the above c, the requester gets a signature c, s for the original message m.

(v) (5) Anyone can verify the signature (c, s) by checking the following equation:

$$c = \mathrm{SHA256}\left(m \| R_x(cP + sG) \bmod n\right). \tag{3}$$

Note that here $R_x(cP + sG) \bmod n$ means we get the x resolution values of point $cP + sG$, and || means we concatenate two strings.

3.3. Blockchain. Maintained by many mutual untrusted parties, the leger of a blockchain generally captures the characteristics of decentralization, tamper proof, and traceability. Blockchain technology is an underlying technology of the famous cryptocurrency Bitcoin [30] and has been a prominent development in the past decade. Consequently, many applications are built based on blockchain to acquire the characteristics, so as our proposed voting protocol in this study. The key notations of blockchain technology are as follows:

(i) Ledger. As the name implies, the ledger is used to manage data such as accounts and transaction flow and supports functions such as classified bookkeeping, account reconciliation, and clearing and settlement. In multiparty cooperation, multiple participants hope to jointly maintain and share a timely, correct, and secure distributed ledger to eliminate information asymmetry, improve operational efficiency, and ensure capital and business security. The blockchain is usually regarded as a core technology for building a "distributed shared ledger." Through the joint of a series of technologies such as chained block data structures, multiparty consensus mechanisms, smart contracts, and world state storage, it can achieve a shared ledger that is consistent, credible, transactionally secure, and difficult to tamper with. The basic contents contained in the ledger include blocks, transactions, accounts, and world states.

(ii) Block. Blocks are data structures constructed in chronological order. The new block will introduce the hash information of the previous block and then use the hash algorithm and the data of this block to generate a unique data fingerprint. The sophisticated data structure design makes the data on chain traceable and verifiable.

(iii) Transaction. A transaction can be regarded as a piece of request data sent to the blockchain system, which can be used to deploy contracts, call contract interfaces, maintain the life cycle of contracts, manage assets, and exchange value. The basic data structure of a transaction includes sender, receiver, and transaction data.

(iv) Consensus Mechanism. The consensus mechanism is a core concept in the blockchain. As a distributed system, the blockchain can be jointly calculated by different nodes, which jointly witness the execution process of transactions and confirm the final

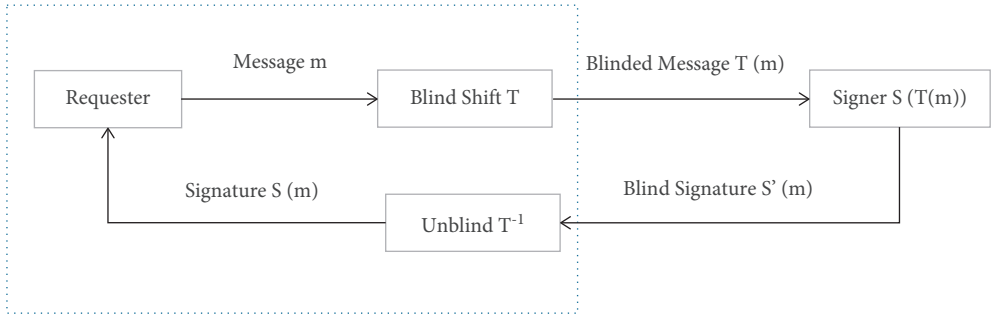


Figure 2: Blind signature scheme.

calculation results. There is a process of cooperating in the blockchain that it can make mutually untrusted participants to reach an agreement and ensure consistency. Continuous cooperation can be abstracted as a "consensus" process. The algorithms and strategies involved are collectively referred to as a consensus mechanism.

(v) Smart Contract. A smart contract refers to a contract defined in digital form that can automatically execute terms. The digital form means that the contract must be implemented in computer code. As long as the parties reach an agreement, the rights and obligations established by the smart contract will be automatically executed. Thus, the result cannot be denied. To run digital smart contracts, the blockchain system must have compilers and executors that can compile, parse, and execute computer code, collectively referred to as a virtual machine system. After the contract is written, it is compiled with a compiler, and a deployment transaction is sent to deploy the contract on the blockchain system. After the deployment transaction consensus is passed, the system assigns a unique address to the contract and saves the binary code of the contract. After the transaction is called, the virtual machine executor loads the code from the contract storage, executes it, and outputs the execution result.

4. Proposed Protocol

In this subsection, we will describe a practical anonymous voting protocol via blockchain. Our scheme has all the aforementioned properties in Section 1.

4.1. System Overview. As shown in Figure 3, the voting protocol consists of four stages: initialization, voting, opening, and verifying/tally. We adopt the ECC system in our scheme, and the elliptic curve is secp256k1. This curve can be described as $T = (p, G, n, a, b, h)$, where a and b are constants, p is the p value of the finite field $F(p)$ of secp256k1, G is the base point, n is the order of G, and h is a cofactor. All these parameters are public.

4.1.1. Initialization. Anyone in the IoE system can launch voting by the proposed IoEPAV. All voters who want to join the voting should provide their public keys and identification. In the initialization stage, all the public information of the voters will be broadcasted to the blockchain through the smart contract in IoEPAV. We assume that there are n_v different voters $v_1, v_2, \ldots, v_{n_v}$. Each voter v_i generates two pairs of ECC keys (sk_{v_i}, pk_{v_i}) and $(\widehat{sk}_{v_i}, \widehat{pk}_{v_i})$. Let addr_{v_i} be the public address of the voter v_i in the Ethereum network, and ID_{v_i} represents the voter's identification. Then, the public information for each voter v_i is a tuple $(ID_{v_i}, pk_{v_i}, \widehat{pk}_{v_i}, \text{addr}_{v_i})$. Everyone can get this information from the blockchain to verify its validity.

4.1.2. Voting. As soon as voting is launched, each voter v_i can start to submit their ballot. Firstly, a cryptography commitment protocol is invoked $\text{Commit} = (C, V)$. Here, we use the algorithm C and the algorithm V will be used later in the final stage. C as $(c_{v_i}, o_{v_i}) \xleftarrow{R} C(m_{v_i})$ is invoked for the ballot message m_{v_i} of voter v_i.

Then, voter v_i generates a $\widetilde{c} = (\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_{n_v})$ and $\widetilde{x} = (\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_{n_v})$ for different voters $v_1, v_2, \ldots, v_{n_v}$, where $v_{n_v} \neq v_i$. Voter v_i completes this by a blind commitment algorithm $(\widetilde{x}, \widetilde{c}) \leftarrow \text{BlindX}(c_{v_i}, \widetilde{pk}_{n_v}, \widetilde{\widetilde{pk}}_{n_v})$ in the blind signature protocol.

Let H_{n_v} be the hash of the tuple $(\text{addr}_i, ID_i, \widehat{c}_{n_v})$ for different voters $v_1, v_2, \ldots, v_{n_v}$, where $v_{n_v} \neq v_i$. Then, voter v_i uses the ECDSA signature algorithm $s_{n_v} = \text{Sign}(sk_i, H_{n_v})$ for other different voters n_v and gets a tuple of signatures $\widetilde{s} = (s_1, s_2, \ldots, s_{n_v})$.

Then, a group information of $(ID_i, \text{addr}_i, \widetilde{c}, \widetilde{s})$ is recorded into the blockchain through the smart contract. Note that $\widetilde{x}$ has been saved in secret by voter v_i himself in this stage. The detailed design of the algorithm C and BlindX is given in Section 4.2.

Once $(ID_i, \text{addr}_i, \widetilde{c}, \widetilde{s})$ generated by voter v_i is recorded, the other voters can generate a blind signature for it. Firstly, every other voter verifies the validity of the signature s_{n_v} by the ECDSA verification algorithm $\text{Verify}(s_{n_v}, pk_i)$. If s_{n_v} is valid, then every other voter generates a blind signature d_{n_v} by a blind signature algorithm $d_{n_v} \leftarrow \text{Blind}S(\widehat{sk}_{n_v}, sk_{n_v})$ in the blind signature protocol and sends d_{n_v} into the blockchain. Let $\widetilde{d} = (d_1, d_2, \ldots, d_{n_v})$ for different voters $v_1, v_2, \ldots, v_{n_v}$ where $v_{n_v} \neq v_i$.

At the end of this stage, we have $(\widetilde{c}, \widetilde{s}, \widetilde{d})$ for a ballot m_{v_i} of voter v_i. The detailed design of the algorithm Blind$S$ will be given in Section 4.2.

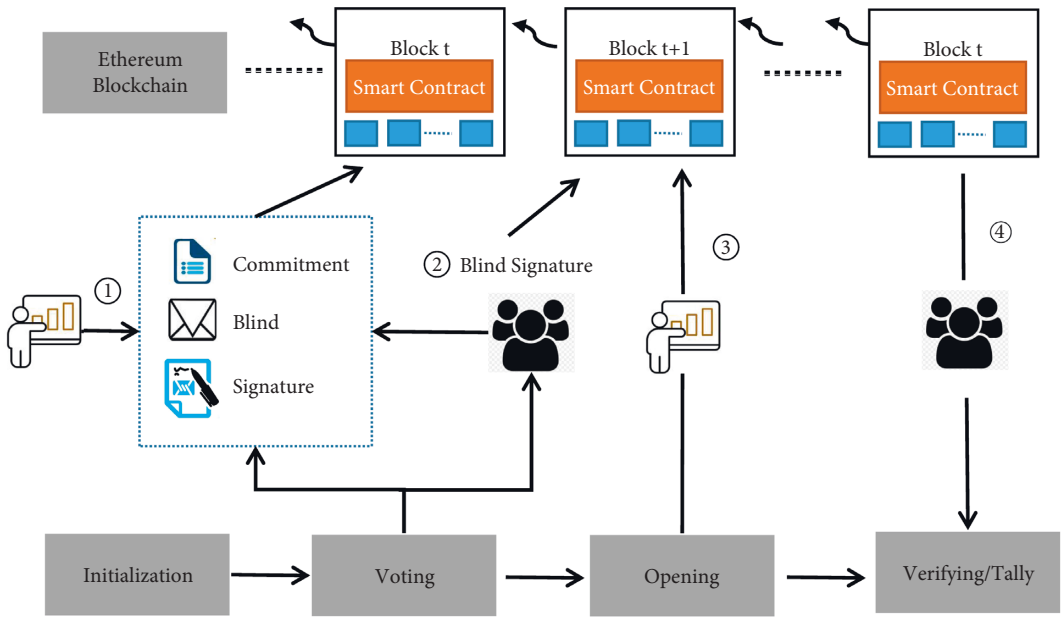


Figure 3: Proposed IoEPAV.

4.1.3. Opening. In this stage, voter v_i will open his/her voting commitment. Firstly, he/she gets the $\tilde{d}$ from the blockchain and calculates the corresponding tuple $\tilde{y} = (y_1, y_2, \ldots, y_{n_v})$ for different voters $v_1, v_2, \ldots, v_{n_v}$ where $v_{n_v} \neq v_i$. Here, y_{n_v} is a signature for the original commitment c_{v_i}. Voter v_i generates $\tilde{y}$ by calculating

$$y_{n_v} = d_{n_v} + \gamma, \tag{4}$$

where γ is a random key saved in Blind$X$.

Given $(x_{n_v}, c_{v_i}, y_{n_v})$, voter v_i can verify the validity of y_{n_v} by an algorithm Verify$S(x_{n_v}, c_{v_i}, y_{n_v}, pk_{n_v})$, where x_{n_v} is saved in Blind$X$. Note that y_{n_v} is a signature of the commitment c_{v_i} from voter v_{n_v}. If the signature y_{n_v} is valid, then voter v_i sends the open commitment string o_{v_i} and x_{n_v} to the blockchain. Note that voter v_i chooses a random address in the Ethereum network to send this transaction. Anyone cannot find out who has sent this transaction. The detailed design of the algorithm Verify$S$ will be given in Section 4.2. Finally, we have $(c_{v_i}, o_{v_i}, \tilde{x}, \tilde{y})$ for an original ballot m_{v_i} of voter v_i in the blockchain.

4.1.4. Verifying/Tally. Now, we have a voting list of $(c_{v_i}, o_{v_i}, \tilde{x}, \tilde{y})$ for each voter v_i. Everyone can use Verify$S$ again to verify the voting. Note that we do not have to verify all the $\tilde{y}$ values for a ballot m_{v_i} of voter v_i. If more than half of $\tilde{y}$ are valid, we think (c_{v_i}, o_{v_i}) is valid. Finally, we can use the $V(c_{v_i}, o_{v_i})$ algorithm in cryptography commitment to open the original ballot m_{v_i} and tally the result of the voting.

4.2. Algorithm Design. In this subsection, we will describe the design of the algorithms mentioned in A in detail. To make the voting service as simple as possible to be integrated into an IoE system, we assume that anyone who tries to use the voting service can ignore the underlying design of smart contracts of the blockchain. The details are as follows:

(i) Cryptography Commitment Algorithms. To construct a cryptography commitment protocol, we should build a pair of efficient algorithms (C, V). We can do this by using a collision-resistant hash function. As shown in Algorithm 1, the algorithm C generates a 32 byte random string as the open string o in the commitment protocol. The commit string c is the hash of the original ballot m and o. Algorithm 2 is a invert process to verify whether (m, o) is corresponded to c. The algorithm C is used at the beginning of the "Voting" stage, while the algorithm V is used at the end of the "Verifying/Tally" stage. We will give the security analysis for this commitment protocol in Section 5.

(ii) Blind Algorithms. We divide the blind signature protocol into three algorithms. They are Blind$X$, Blind$S$, and Verify$S$. As shown in Algorithm 3, Blind$X$ is used for blinding the commit string c_{v_i} for the ballot. γ and δ are two random secret keys. G and n are the public parameters in the ECC and the commit string; the input also includes the public keys of each voter. Following the blind signature protocol mentioned in Section 3, Blind$X$ is used to generate the blinded message $(\tilde{s}, \tilde{c})$. At the end of Blind$X$, each voter keeps the secret data $(x_j, \hat{c}_j, \gamma, \delta)$ and sends $(\text{addr}_{v_i}, \widetilde{\text{addr}}_{n_v}, \tilde{s}, \tilde{c})$ to the blockchain.

Then, in Algorithm 4, Blind$S$ is used for the other voters to sign on $\hat{c}_{n_v}$ submitted by voter v_i in Algorithm 3. With the corresponding secret keys $(\hat{s}k_{n_v}, sk_{n_v})$, each signer can generate a blind signature d_{n_v} as Line 8 in Algorithm 4. Then, voter v_i can easily calculate the explicit signature $y_{n_v} = d_{n_v} + \gamma$ for c_{v_i}. Finally, we have Verify$S$ as shown in

Input (m)
Output (c,o)
(1) o = crypto.randomBytes(32)
(2) c = SHA256(m + serialize(o))
(3) **return** (c, o)

Algorithm 1: Generate commitment.

Input (m, c, o)
Output *True* or *False*
(1) $\hat{c}$ = SHA256(m + serialize(o));
(2) **if** $c \neq \hat{c}$ **then**
(3) **return** *False*
(4) **else**
(5) **return** *True*
(6) **end if**

Algorithm 2: Open commitment.

Input $(c_{v_i}, \widetilde{\widetilde{pk}_{n_v}}, \widetilde{pk}_{n_v})$
(1) γ = crypto.randomBytes(32);
(2) δ = crypto.randomBytes(32);
(3) **for** $j = 0$ to n_v **do**
(4) $A_j = \widehat{pk}_j + \gamma * G + pk_j * \delta$
(5) t_j = getXpointFromPubkey (A_j)mod n
(6) x_j = SHA256($c_{v_i} + t_j$)
(7) $\hat{c}_j = x_j - \delta$
(8) $\tilde{c}$.push$(\hat{c}_j)$
(9) H_i = SHA256 $(ID_i + \text{addr}_i + \hat{c}_j))$
(10) s_j = secp256k1.ecdsaSign(H_i, sk_i)
(11) $\tilde{s}$.push(s_j)
(12) save(x_j, $\hat{c}_j$, γ, δ)
(13) **end for**
(14) await Contract.setAnmVote(addr$_{v_i}$, $\widetilde{\text{addr}}_{n_v}$, $\tilde{s}$, $\tilde{c}$);

Algorithm 3: BlindX.

Algorithm 5 to verify the validity of the signature y_{n_v}. Everyone who gets the public key of the signers from the blockchain can calculate the hash. Here, the blind signature protocol mentioned in Section III is divided into Algorithms 3 to 5.

4.3. Smart Contract Design. In this part, we present the design of the smart contract, which provides interfaces to record the voting data into the blockchain. Thus, the smart contract should involve the necessary data structure referring to the voting scheme. Firstly, we need a data structure to record the information binding with the voter. As shown in Table 2, "address, PK, PKs, and ID" are the basic public information, while the other three mapping data are corresponding to process data generated in the blind signature protocol. Table 3 is designed for storing the results in the "opening" stage. Besides, there are some other variables such as an "unit" for the number of voters. To be succinct, we do not list all of them.

Recall that there are four stages in the whole voting scheme, namely, initialization, voting, opening, and verifying/tally. Then, the smart contract should afford the necessary interfaces for them to interact with the blockchain. The interfaces can be classified into two types: "Write" for recording information into the blockchain and "Query" for querying information from the blockchain. We have 11 interfaces in our smart contract. Considering the space of the paper, only those critical functions are given in detail. However, it is enough for the readers to understand the whole protocol. In the stage "Initialization," a kind of "Write" function is used to record the public identity information of a voter. Then, in the stage "Voting," a "Write" interface named "setAnmVote" is used to record the blind commitment generated in Algorithm 3. As shown in Algorithm 6, *iV* and *iS* are the accounts in the Ethereum blockchain that represents a voter's address and a potential signer's address separately. *isi* and *ici_pi* are data generated in "BlindX." voters is an array corresponding to the data structure in Table 2. The requirement in Line 1 makes sure that only the voter himself can set the data. Once the data have been recorded, no one can reset it including the voter himself.

As soon as the data are confirmed by the blockchain, the other voters acting as a signer will try to generate a blind signature for the commitment. The algorithm for a signer to generate a blind signature is described in Algorithm 4. Firstly, a signer will use "getAnmVote" in Algorithm 7 to query the commitment data generated for him. Anyone can query the commitment according to the public Ethereum account address. In the end, the signer will submit his blind signature idsig through "signAnmVote" in Algorithm 8. Similarly, the requirement in Line 1 makes sure that only the signer himself can set the corresponding data. Once the signature has been recorded, no one can reset it including the signer himself.

The design ideas of the other interfaces are similar to these algorithms given above. When "Write" information to the blockchain, necessary conditions are set. Then, anyone can check the data from the blockchain by the kind of "Query" interface.

5. Security Analysis

In this section, we will discuss why our protocol can resist the potential attacks in the threat models and fulfil all the design goals in Section 3.

5.1. Threat Models. In our scheme, we suppose a voter is a rational one, which means he/she would not let their right to vote become invalid by doing something obviously break the protocol. For instance, a voter should submit his/her blind signature for other voters' ballots correctly; otherwise, his/her right to vote will be thought invalid. Here, we present the threat model specially for a voting service.

Input $(\widehat{sk}_{n_v}, sk_{n_v})$
(1) (v, s_{n_v}, $\widehat{c}_{n_v}$) = await Contract.getAnmtVote (addr$_i$, addr$_{n_v}$)
(2) **if** $v \neq$ addr$_i$ **then**
(3) Return
(4) **end if**
(5) H = SHA256 (ID_i + addr$_i$ + $\widehat{c}_{n_v}$)
(6) S = secp256k1.ecdsaVerify(s_{n_v}, H, pk_i)
(7) **if** S == True **then**
(8) $d_{n_v} = \widehat{sk}_{n_v} - \widehat{c}_{n_v} * sk_{n_v}$
(9) **else**
(10) **Return**
(11) **end if**
(12) await Contract.signAnmVote(addr$_i$, d_{n_v})

Algorithm 4: BlindS.

Input $(x_{n_v}, c_{v_i}, y_{n_v}, pk_{n_v})$
Output *True* or *False*
(1) $B = x_{n_v} * pk_{n_v} + y_{n_v} *$ G
(2) bx = getXpointFromPubkey(B) mod n
(3) H = SHA256(c_{v_i} + bx)
(4) **if** $H == x_{n_v}$ **then**
(5) **Return** *True*
(6) **else**
(7) **Return** *False*
(8) **end if**

Algorithm 5: VerifyS.

Input $(iV, \widetilde{\text{addr}}, \widetilde{s}, \widetilde{c})$
(1) require(msg.sender == iV);
(2) **for** $i = 0$ to n_v **do**
(3) $iS = \widetilde{\text{addr}}$ [i];
(4) $isi = \widetilde{s}$ [i];
(5) $ici\_pi = \widetilde{c}$ [i];
(6) require(voters [iV].si [iS]) == 0);
(7) voters [iV].si [iS] = isi;
(8) voters [iV].ci_pi [iS] = ici_pi;
(9) **end for**

Algorithm 6: setAnmVote.

Table 2: Structure of a voter.

Data type	Description
address	voter
string	PK
string	PKs
string	ID
mapping(address ⇒ string)	si
mapping(address ⇒ string)	ci_pi
mapping(address ⇒ string)	dsigs

Table 3: Structure of an open result.

Data type	Description
string	m
string	bm
string	oi
mapping(address ⇒ string)	ci
mapping(address ⇒ string)	yis
bool	isFinished

(1) Voter Model. Although a voter is a rational one, he/she may try to lead the voting result in his/her favour without breaking the rule of the protocol. First, since the voting is anonymous, a voter may attempt to submit a duplicate ballot to increase his/her chance to vote. Second, because each voter will blindly sign on the blinded ballots, it is possible for a voter to attempt to change the original ballot after the corresponding blind ballot has been blindly signed by others. It means any vulnerability of the blind signature protocol will defeat the whole voting scheme. Third, knowing other voters' public identities, a voter may look for ways to let others' legal ballots become invalid by forging others' ballots.

(2) Adversary Model. An adversary can be anyone who is a user of the IoE system. First, an adversary may attempt to affect other voters' choices through vote buying, voter coercion, and so on. Second, there is a possibility for an adversary to stop an eligible voter from performing the process of the voting protocol. For example, voters can be subjected to DDoS attacks, causing them to malfunction. Third, an adversary may attempt to tamper with the result of the voting.

(3) Blockchain System Model. Attacks against the blockchain system may also cause the failure of the voting scheme, since it is based on the blockchain.

5.2. Cryptography Commitment. A cryptographic commitment scheme Commit = (C, V) is secure when it is both hiding and binding. In our scheme, we constructed the cryptographic commitment using a collision-resistant hash function H (in out construction, we use SHA256).

Input (iV, iS)
Output (V, S, isi, ici_pi)
(1) return (voters $[iV]$.voter, voters $[iV]$.si $[iS]$, users $[iV]$.ci_pi $[iS]$);

Algorithm 7: getAnmVote.

Input (iV, idsig)
(1) require(voters $[iV]$.voter.isvaild() and users $[iV]$.dsigs[msg.sender]).length = = 0);
(2) voters $[iV]$.dsigs[msg.sender] = idsig;

Algorithm 8: signAnmVote.

We now prove that the binding commitment C_H satisfies two properties based on the assumption that H is collision-resistant.

(i) Binding Proof. The binding commitment C_H is a binding commitment if H is collision-resistant. This can be shown immediately as follows: if there exists an adversary A that breaks the binding property, it will immediately give a collision for H. More precisely, for some commitment string c, assume A outputs two pairs (m_1, o_1) and (m_2, o_2), where $m_1 \neq m_2$, but $V(m_1, c, o_1) = v(m_2, c, o_2) = \text{accept}$. Thereafter, we have a collision for H that $H(h_1, o_1) = c = H(m_2, o_2)$. So, we can say that C_H is computationally binding since it depends on a computational assumption for solving this collision for H.

(ii) Hiding Proof. We first consider input hiding required that the distribution $\{H(m_1, o)\}$ is statistically indistinguishable from the distribution $\{H(m_2, o)\}$ for all $m_1, m_2 \in M$, where $o \xleftarrow{r} R$. In our construction, once H is collision-resistant and if the set R is large enough, it is considered input hiding. For example, $R = \{0,1\}^{512}$ should be sufficient for SHA256. This provides a way to build a secure and practical commitment scheme from SHA256. Then, if H is input hiding, no adversary even an unbounded adversary A can break the security of its derived commitment scheme C_H. So, we can say that C_H is unconditionally hiding.

5.3. Blind Signature. The blind signature protocol we used in our scheme is recalled as follows:

(i) (1) The signer n_v randomly generates two pair keys (sk_{n_v}, pk_{n_v}) and $(\widehat{sk}_{n_v}, \widehat{pk}_{n_v})$ and $pk_{n_v}, \widehat{pk}_{n_v}$ are public to the requester v_i.

(ii) (2) The requester selects two integers γ and $\delta \in \mathbb{Z}_n$ randomly and computes $A = \widehat{pk}_{n_v} + \gamma * G + \delta * pk_{n_v} = (xp, yp), t = xp \bmod n$. It checks whether t equals zero. If so, the requester reselects γ and δ. Then, it computes $x = \text{SHA256}(c_{v_i} \| t)$ and $\widehat{c} = x - \delta$, where SHA256 is a hash function with 32 bit words and c_{v_i} is the commitment generated by voter v_i. Finally, the requester sends $\widehat{c}$ to the signer as the blinded message.

(iii) (3) The signer generates a blind signature $d = \widehat{sk}_{n_v} - \widehat{c} * sk_{n_v}$ using the blinded message and sends to the requester.

(iv) (4) On receiving d, the requester computes $y = d + \gamma$, and with its above x, the requester gets a signature x, y for the original message c_{v_i}.

(v) (5) Anyone can verify the signature (x, y) by checking the following equation: $x = \text{SHA256}(c_{v_i} \| R_x(x * pk_{n_v} + yG) \bmod n)$.

Correctness Proof. Firstly, we prove the signature (x, y) to be valid as follows:

$$\begin{aligned} & R_x\left(x * pk_{n_v} + yG\right) \bmod n, \\ & = R_x\left(x * pk_{n_v} + d * G + \gamma * G\right) \bmod n, \\ & = R_x\left(x * pk_{n_v} + \widehat{sk}_{n_v} * G - \widehat{c} * sk_{n_v} * G + \gamma * G\right) \bmod n, \\ & = R_x\left(x * pk_{n_v} + \widehat{sk}_{n_v} * G - (x - \delta) * sk_{n_v} * G + \gamma * G\right) \bmod n, \\ & = R_x\left(\widehat{pk}_{n_v} + \gamma * G + \delta * pk_{n_v}\right) \bmod n, \\ & = R_x(xp, yp) \bmod n, \\ & = t. \end{aligned} \tag{5}$$

It follows that $x = \text{SHA256}(c_{v_i} \| R_x(x * pk_{n_v} + yG) \bmod n)$, which means that (x, y) is a valid signature of c_{v_i}.

Blindness Proof. Secondly, we show the blindness of the protocol. We define view $\mathbb{V}$ for a signer during the process of the protocol. For example, let (x, y) be the signature of c_{v_i} that has been generated in the protocol. Then, view $\mathbb{V}$ consists of sk_{n_v}, $pk_{n_v} = sk_{n_v} * G$, $\widehat{sk}_{n_v}$, $\widehat{pk}_{n_v} = \widehat{sk}_{n_v} * G$, $\widehat{c}$, and $d = \widehat{sk}_{n_v} - \widehat{c} * sk_{n_v}$. We then show that for any given view $\mathbb{V}$ and valid message signature pair $(c_{v_i}, (x, y))$, blinding factors γ and δ exist and are unique. Then, for γ and δ, we have

$$\begin{aligned}\widehat{c} &= (x-\delta)\mathrm{mod}n,\\ \gamma &= \left(y-\widehat{sk}_{n_v}+\widehat{c}*sk_{n_v}\right)\mathrm{mod}n,\\ x &= \mathrm{SHA256}\left(c_{v_i}\middle\|R_x\left(\widehat{pk}_{n_v}+\gamma*G+\delta*pk_{n_v}\right)\mathrm{mod}n\right).\end{aligned} \tag{6}$$

Then, $R_x(\widehat{pk}_{n_v}+\gamma*G+\delta*pk_{n_v})\mathrm{mod}n$ is uniquely determined by x and c_{v_i}. To make it succinct, we note $\widehat{sk}_{n_v}+\gamma+\delta*sk_{n_v}=R_x^{-1}(c_{v_i}\mathrm{°x})$, which means $\widehat{sk}_{n_v}+\gamma*G+\delta*sk_{n_v}$ is uniquely determined by x and c_{v_i}.

Then, $\delta*sk_{n_v}=(R_x^{-1}(c_{v_i}\mathrm{°x})-\widehat{sk}_{n_v}-\gamma)\mathrm{mod}n=(R_x^{-1}(c_{v_i}\mathrm{°x})-\widehat{\mathrm{sk}}_{\mathrm{n_v}}-\mathrm{y}+\widehat{\mathrm{sk}}_{\mathrm{n_v}}-\widehat{c}*\mathrm{sk}_{\mathrm{n_v}})\mathrm{modn}=(\mathrm{R}_{\mathrm{x}}^{-1}(\mathrm{c}_{\mathrm{v_i}}\mathrm{°x})-\mathrm{y}-\widehat{c}*sk_{n_v})\mathrm{mod}n$.

Finally, we have

$$\begin{aligned}\gamma &= \left(y-\widehat{sk}_{n_v}+\widehat{c}*sk_{n_v}\right)\mathrm{mod}n,\\ \delta*sk_{n_v} &= \left(R_x^{-1}\left(c_{v_i}\mathrm{°x}\right)-y-\widehat{c}*sk_{n_v}\right)\mathrm{mod}n.\end{aligned} \tag{7}$$

Then, γ and δ can be uniquely determined by $(x, y, c_{v_i}, \widehat{sk}_{n_v}, sk_{n_v}, \widehat{c})$. All $(x, y, c_{v_i}, \widehat{sk}_{n_v}, sk_{n_v}, \widehat{c})$ are in the view $\mathbb{V}$.

5.4. Security Properties. First, we will show how the protocol fulfils the seven design goals.

Fairness. The proposed voting scheme is a fair voting by succeeding the hiding property of the cryptography commitment protocol. Since we have proven the security of the cryptography commitment protocol used in our scheme, no one can obtain the ballot results of others before he/she has submitted his/her ballot.

Decentralization. Directly, the protocol is decentralized as it is built without any TTP. We use smart contracts to automate the voting process of the Internet of energy. Eligibility. In the initialization stage, we check all the voters' identities and public information to make sure the eligibility.

Anonymity. The correctness and blindness of the blind signature protocol make sure that no one can know the owner of a ballot from the voting result at the end. Since we have proven the security of the blind signature protocol used in our scheme, the privacy of voters can be protected.

Compatibility. As the protocol is realized by smart contract and Web3 [31], it can be integrated into an IoE system easily. Any blockchain system including Ethereum 2.0, which supports smart contracts, is feasible for the proposed scheme and we do not need to construct a whole new blockchain platform.

Verifiability. The data generated in each stage of the voting can be checked from the blockchain. Therefore, the voting is verifiable.

Coercion Resistance. Even though one tries to compel a voter to vote by his/her instruction, he/she cannot find out whether the coerced voter has done as he/she wishes.

5.5. Resistance against the Threat Model. Finally, we will show how the protocol resists the potential attacks in the threat models. (1) Resistance against Voter: first, a voter cannot submit a duplicate ballot because the design of the smart contract will reject a piece of duplicate information. Second, a voter cannot change the original ballot for the security of the blind signature protocol. Third, to forge others' ballots, a voter should get their secret keys or break the ECC asymmetric cryptography. (2) Resistance against Adversary: first, since the voting is anonymous, an adversary cannot affect other voters' choices. Second, to stop an eligible voter from voting, an adversary should break the security of the blockchain. The blockchain makes sure that the result of the voting cannot be tampered with.

(3) Resistance against Blockchain: because our protocol is designed via the Ethereum blockchain, which is proven secure in practice, and thousands of applications have been built on it. A 51% attack is still hypothetical by a group of miners controlling more than 50% of the network's mining hash rate or computing power. For voting, if more than 50% of voters collude, it is not necessary to launch voting. Besides, the fully decentralized architecture of blockchains makes them robust against DoS/DDoS attacks.

6. Implementation and Performance

6.1. Implementation Description. We use a PC with an OS version of Ubuntu 18.04 64x as a user client. The CPU and memory are Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz 2.30 GHz and 8G separately. We implemented our protocol in two parts, namely, Web3 programs and smart contracts. We write the smart contract in Solidity. We use JavaScript to finish the Web3 programs along with several libraries. "Ethers.js" is an Ethereum library to interact with the Ethereum blockchain. "Crypto.js" is a JavaScript library to realize the cryptography protocol adopted in our voting protocol. The experiment code is available at https://github.com/researchSec/IoEPAV. As shown in Figure 4, each user plays as a voter and a blind signer simultaneously. They interact with the blockchain network through the Web3 application. Indeed, the Web3 application will invoke the smart contract to read or write voting data on the blockchain network. Note that the users do not need to involve in the mining or consensus progress of the blockchain network. For simplicity, we use multiple Ethereum accounts to represent different voters in the Web3 application to invoke the smart contract. To evaluate the performance in a more reliable and practical manner, we deploy the smart contract on two popular test networks of Ethereum and have strong links to the main network. Besides, we build an Ethereum development with a professional tool called Hardhat. Thus, in the connectivity model, we run our test cases in three blockchain networks separately. They are Hardhat local network, Rinkeby test network, and Ropsten test network.

The Ethereum network that deals with real money is called "mainnet," and then, other live networks named "testnets" (multiple ones) are also provided by Ethereum. In testnets, the network does not deal with real money but does mimic the real-world scenario well. Ropsten and Rinkeby are the testnets we choose. Ropsten is a proof of work (PoW) testnet. This means it is the best like-for-like representation of Ethereum. Rinkeby is proof of authority (PoA) [32] testnet. In Table 4, we give the state of the two blockchain networks when we performed the experiment. In the testnet

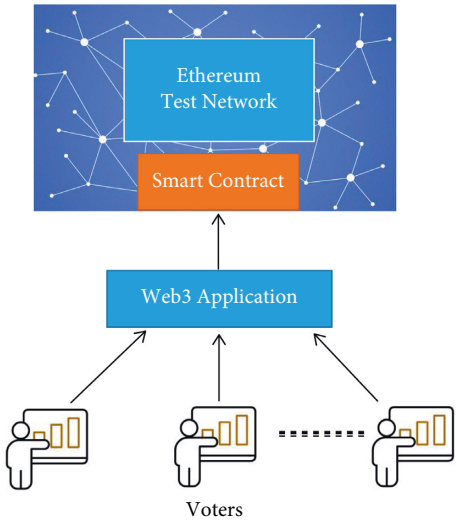


Figure 4: Connectivity model.

Table 4: State of the blockchain Network.

Test network	Consensus	Average block time (S)	Active nodes/ miners	Latest block number
Rinkeby	PoW	47.18	42	10027471
Ropsten	PoA	15.03	44	11869370

Ropsten, the coin is mined following the same scheme as the mainnet. Rinkeby uses PoA consensus schemes, which are a potential direction of Ethereum evolutions.

6.2. Performance in Local Environment. Firstly, we test the key steps in the local development environment to see the performance without considering the latency and throughput according to the network condition. As shown in Table 5, the time for each stage of the voting is proportionate to the increased number of voters. The result is quite straightforward to understand since more voters mean more commitments to be generated. However, even if the number of voters is up to 50, the time is not more than 2 seconds. The time for verifying does not exceed one second. Thus, one can launch voting efficiently.

6.3. Performance in Live Networks. The performance of the voting scheme is relative to the network condition of the live networks. To give an overall evaluation of the system, we give the average response time, transaction cost, throughput, and latency for the execution of the smart contract. To make it clear, we give the description of evaluation properties as follows.

Average Response Time. This metric indicates the time taken to send a transaction to the blockchain and get a response. Note that when a user gets a response, it does not mean that the sent transaction has been confirmed by the nodes/miners of the blockchain.

Transaction Cost. Gas cost is the transaction cost. Gas refers to the fee that is required to successfully execute a contract on the Ethereum blockchain platform. We use the base unit "Gas." 4×10^{-9} Ether $\approx$ 1 Gas.

Transaction Acceptance Latency (TAL). Firstly, a signed transaction is created. The user sends the created transaction to an Ethereum network and captures the current time point T_s. When the transaction is confirmed, the user captures the current time T_e. This metric indicates the time of $t = T_e - T_s$.

Throughput. We consider two metrics, namely, (a) read throughput called QPS (query rate per second) and (2) transaction throughput called TPS (transactions per second). QPS indicates the total number of reading transactions performed within a defined period of time. TPS indicates the ratio of valid transactions that are initiated within a defined period of time.

Firstly, we give the response time of each stage in the voting scheme on different test networks. As shown in Figures 5–9, the performances are quite stable in all live networks. The response time for each stage of the voting is proportionate to the increased number of voters. The result is quite straightforward to understand since more voters mean more commitments to be generated. The performances in the two test networks are quite similar.

To evaluate how the network conditions affect the performance of the voting scheme, we give the transaction acceptance latency (TAL) and throughput in Figures 10 and 11. The TPS and QPS are quite similar between Rinkeby and Ropsten. This can be an explanation for the result of the response time above. Since the throughput in the mainnet of the Ethereum blockchain is about 85 times the size of that in the test networks, we can infer that the performance of the voting scheme will act more efficiently in the mainnet. Note that the TAL varies significantly with different loads (different numbers of concurrent transactions). In the experiment, we give the number of concurrent transactions according to the number of voters.

In Table 6, we also give the average transaction fee for the smart contract deployment and each stage of the voting scheme. As we can see, the gas cost in the two test networks has little difference since the size of the smart contract and transactions are identical.

6.4. Cost Comparison. In Yang's blockchain-based scheme [16], basic cryptographic operations are also introduced. The difference is we use scalar multiplication on an elliptic curve, while Yang uses exponentiation in a multiplicative group. However, if we only care about computational complexity, it is similar.

(i) In ECC, given an elliptic curve of size n, the number of double-and-add steps is proportional to $O(k)$ for $k * G$. Each double/add is a sequence of a constant number of field multiplications, squares, additions, and subtractions. Multiplication and squares are the expensive ones, and using the Karatsuba algorithm as mentioned in [33], they are $O(n^{1.58})$. Therefore, the result is $O(kn^{1.58})$.

(ii) In comparison, the computational complexity of a modular exponentiation of form $g^a \bmod b$ is similar.

Table 5: Average response time.

Number of voters	Initialization (MS)	Blind messages (MS)	Blind signature (MS)	Opening (MS)	Verifying (MS)
5	374.0	323.43	212.0	282.43	109.43
10	635.85	499.14	436.43	561.43	149.43
30	1314.14	1298.71	1234.71	1129.29	228.14
50	2162.28	2314.43	2271.14	1897.43	383.86

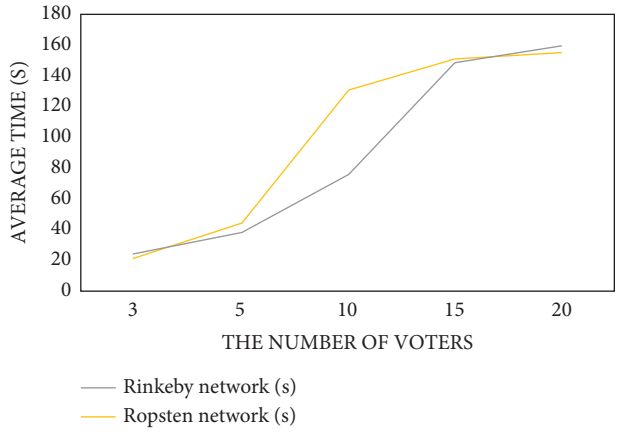


Figure 5: Initialization.

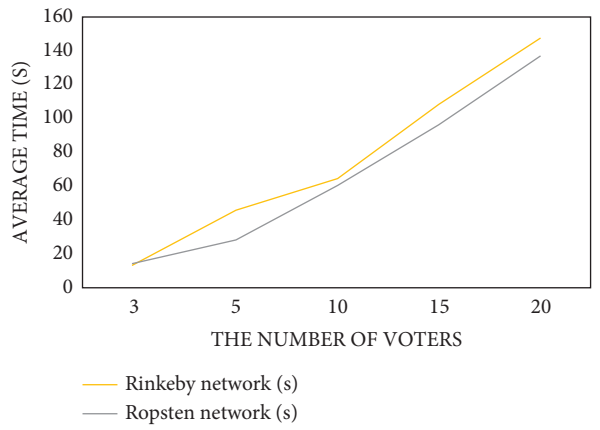


Figure 6: Blind messages.

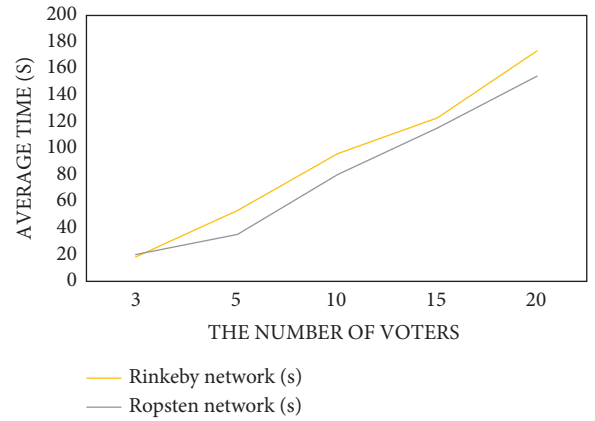


Figure 7: Blind signature.

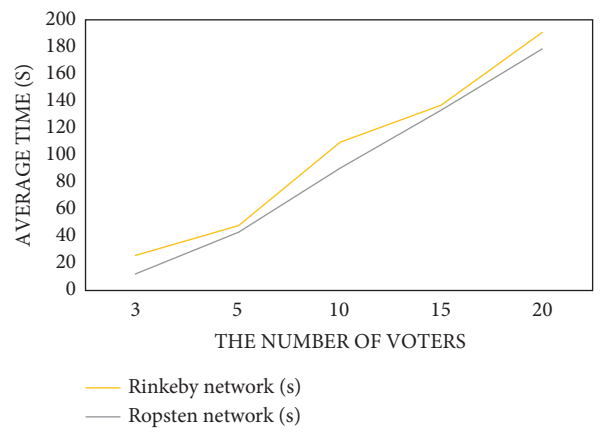


Figure 8: Opening.

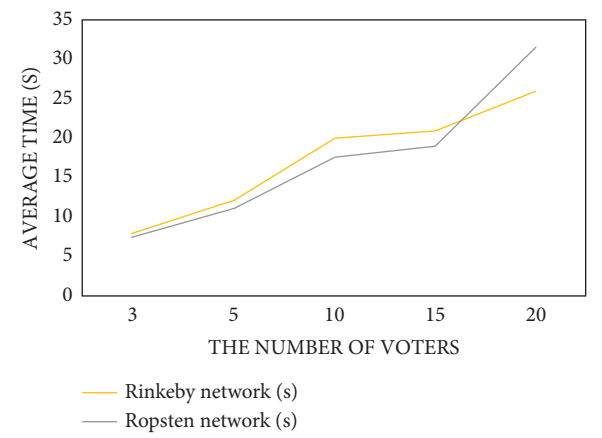


Figure 9: Verifying.

Figure 10: Throughput.

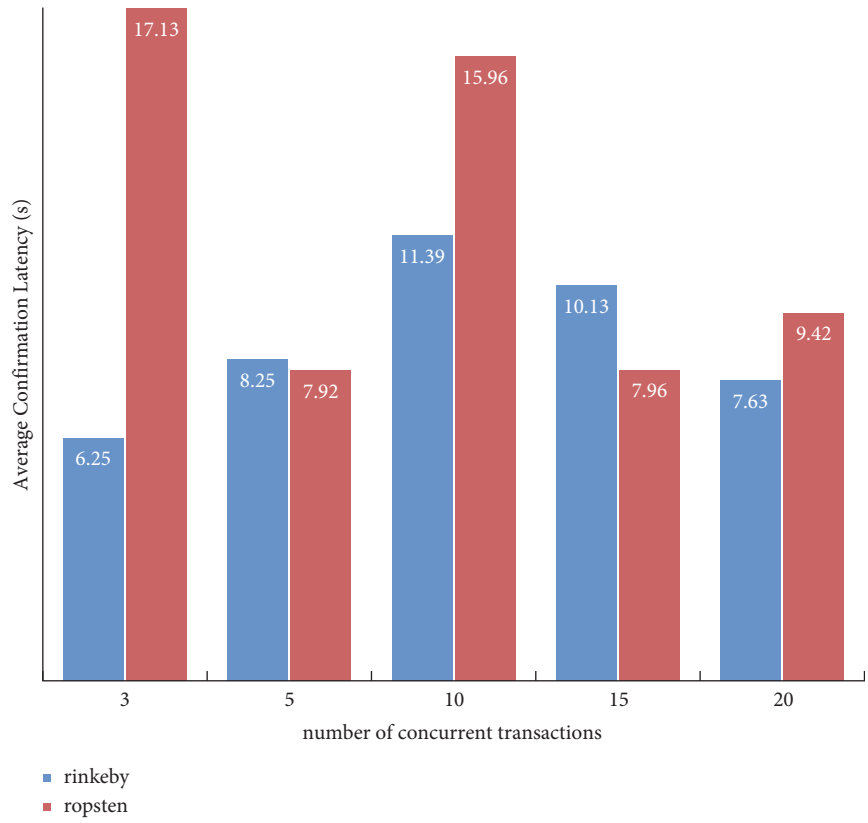


Figure 11: Transaction acceptance latency.

Table 6: Average transaction cost.

Test network	Deployment (Gas)	Initialization (Gas)	Blind messages (Gas)	Blind signature (Gas)	Opening (Gas)
Rinkeby	2,353,838	251,023	204,768	92,811	289,189
Ropsten	2,353,838	251,035	204,780	92,823	284,258

Table 7: Count of the cryptographic operations.

Protocol	Voting	Verifying/tally
IoEPAV	$6 * t * n_v$	$2 * t * n_v$
Yang [16]	$(5 * n_v * t + 3 * t) * n_v$	$(5 * n_v * t + 5 * t) * n_v * n_v$

Square-and-multiply is $O(a)$. Each square/multiply is $O(b^{1.58})$. Therefore, the result is $O(ab^{1.58})$.

According to Yang's performance analysis, only the most time-consuming operations are taken into account. t is denoted as the time of one exponentiation calculation such as a $g^a \bmod b$. Correspondingly, we use t to denote the time of one multiplication calculation in ECC such as $k * G$. Let t_E and t_D be the time of encryption and decryption separately. Then, we let t_S and t_V be the time of executing ECDSA signature and verification, respectively. Approximately, $t_E = 2t$, $t_D = t$, $t_S = t$, and $t_V = t$. n_v is used to denote the number of voters. In Yang's scheme, n_c is the number of candidates, respectively. Note that we can take over the blind signer to candidates in our protocol. To be succinct for comparison, we let $p = n_c = n_v$. The cost of these operations in Yang's and our scheme for comparison is given in Table 7. As shown in Table 7, our scheme IoEPAV is at a lower cost.

7. Conclusion and Future Work

We present a novel blockchain-based voting scheme for IoE system. By getting rid of a trusted third party, the proposed scheme can avoid the single point of failure and is available for a trustless environment. In the past proposals, it is difficult to capture all the required features for a voting scheme. To the best of our knowledge, our scheme is the first one to fulfil all the design goals simultaneously. To achieve this, we combine the cryptography commitment and blind signature protocol. We also use smart contracts to automate the voting process of the Internet of energy. With smart contracts, the voting scheme can be easy to integrate into the IoE system. A voter can follow the voting protocol by invoking the interfaces of the smart contracts. Although we do not use any high-performance cryptography library, the performance in a real environment demonstrates the feasibility of our protocol.

In the future, we can try to use parallel computation in the voting stage to improve efficiency as well. Another

interesting open problem is to create a version of the voting scheme that reduces the number of blind signatures. Making the number of blind signatures irrelative to the number of voters will be a great improvement. This would give a solution that is more efficient with large-scale voters.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partially supported by National Key Research and Development Program of China (grant no. 2019YFE0118700), Science and Technology Project of China Southern Power Grid Corporation (grant no. 066600KK52200016).

References

[1] X. Li, M. Ye, J. Chen, J. Chen, and Y.-C. Chen, "A Novel Hierarchical Key Assignment Scheme for Data Access Control in Iot," *Security and Communication Networks*, vol. 2021, Article ID 6174506, 12 pages, 2021.

[2] B. Xu, Q. Chen, J. Ma, Yu Yan, and Z. Zhang, "Research on a kind of ubiquitous power internet of things system for strong smart power grid," in *Proceedings of the IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, pp. 2805–2808, IEEE, Chengdu, China, May 2019.

[3] Internet of energy for electric mobility home page, http://www.artemis-ioe.eu/ioe_project.htm, 2021.

[4] Y. R. Kafle, K. Mahmud, S. Morsalin, and G. E. Town, "Towards an internet of energy," in *Proceedings of the IEEE International Conference on Power System Technology*, September 2016.

[5] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: a survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[6] J. Yin, Y. Xiao, Q. Pei et al., "Smartdid: a novel privacy-preserving identity based on blockchain for iot," *IEEE Internet of Things Journal*, p. 1, 2022.

[7] T. M. Harrison, T. A. Pardo, and M. Cook, "Creating open government ecosystems: a research and development agenda," *Future Internet*, vol. 4, no. 4, pp. 900–928, 2012.

[8] D. Boneh and V. Shoup, "A graduate course in applied cryptography," 2017, https://crypto.stanford.edu/dabo/cryptobook/BonehShoup04.pdf.

[9] K.-H. Wang, S. K. Mondal, Ki Chan, and X. Xie, "A review of contemporary e-voting: requirements, technology, systems and usability," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 31–47, 2017.

[10] A. Ben, "Helios: web-based open-audit voting," *USENIX security symposium*, vol. 17, pp. 335–348, 2008.

[11] M. Backes, M. Gagné, and M. Skoruppa, "Using mobile device communication to strengthen e-voting protocols," in *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, pp. 237–242, New York, NY, USA, November 2013.

[12] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," pp. 244–251, Springer, Berlin, Germany, 1992.

[13] M. Amine Ferrag and L. Shu, "The Performance Evaluation of Blockchain-Based Security and Privacy Systems for the Internet of Things: A Tutorial," *IEEE Internet of Things Journal*, vol. 8, no. 24, 2021.

[14] FD Team, "Introducing a secure and transparent online voting solution for modern age: follow my vote," https://followmyvote.com.

[15] N. Gailly, P. Jovanovic, B. Ford, J. Lukasiewicz, and L. Gammar, "Agora: Bringing Our Voting Systems into the 21st century," 2018, https://static1.squarespace.com/static/5b0be2f4e2ccd12e7e8a9be9/t/5f37eed8cedac41642edb534/1597501378925/Agora_Whitepaper.pdf.

[16] X. Yang, X. Yi, S. Nepal, A. Kelarev, and F. Han, "Blockchain voting: publicly verifiable online voting protocol without trusted tallying authorities," *Future Generation Computer Systems*, vol. 112, pp. 859–874, 2020.

[17] K. Peng, M. Li, H. Huang, C. Wang, S. Wan, and K. K. R Choo, "Security challenges and opportunities for smart contracts in internet of things: a survey," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12004–12020, 2021.

[18] Q. Pei, E. Zhou, Y. Xiao, D. Zhang, and D. Zhao, "An efficient query scheme for hybrid storage blockchains based on merkle semantic trie," in *Proceedings of the 2020 International Symposium on Reliable Distributed Systems (SRDS)*, pp. 51–60, IEEE, Shanghai, China, September 2020.

[19] M. Ohkubo, F. Miura, M. Abe, A. Fujioka, and T. Okamoto, *An improvement on a practical secret voting scheme*, pp. 225–234, Springer, Berlin, Germany, 1999.

[20] P. Grontas, A. Pagourtzis, and A. Zacharakis, "Coercion resistance in a practical secret voting scheme for large scale elections," in *Proceedings of the 201714th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC)*, pp. 514–519, IEEE, Exeter, UK, June 2017.

[21] G. Srivastava, A. Dhar Dwivedi, and R. Singh, "Phantom protocol as the new crypto-democracy," in *Proceedings of the 2018IFIP International Conference on Computer Information Systems and Industrial Management*, pp. 499–509, Springer, Olomouc, Czech Republic, September 2018.

[22] C. Braghin, S. Cimato, S. Raimondi Cominesi, E. Damiani, and L. Mauri, "Towards blockchain-based e-voting systems," in *Proceedings of the 2019 International Conference on Business Information Systems*, pp. 274–286, Chittagong, Bangladesh, October 2019.

[23] Li Peng and J. Lai, "Lat-voting: traceable anonymous e-voting on blockchain," in *Proceedings of the 2019 International Conference on Network and System Security*, pp. 234–254, December 2019.

[24] M. Naor, "Bit commitment using pseudorandomness," *Journal of Cryptology*, vol. 4, no. 2, pp. 151–158, 1991.

[25] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*, pp. 199–203, Springer, Berline, Germany, 1983.

[26] Q. C. ShenTu and J. P. Yu, "A Blind-Mixing Scheme for Bitcoin Based on an Elliptic Curve Cryptography Blind Digital Signature Algorithm," 2015, https://arxiv.org/abs/1510.05833.

[27] F. Zhang, C. Wang, and Y. Wang, "Digital signature and blind signature based on elliptic curve," *JOURNAL-CHINA*

INSTITUTE OF COMMUNICATIONS, vol. 22, no. 8, pp. 22–28, 2001.

[28] R. Shaikh, M. Nenova, G. Iliev, and Z. Valkova-Jarvis, "Analysis of standard elliptic curves for the implementation of elliptic curve cryptography in resource-constrained e-commerce applications," in *Proceedings of the 2017 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)*, pp. 1–4, IEEE, Tel-Aviv, Israel, November 2017.

[29] D. Rachmawati, J. T. Tarigan, and A. B. C. Ginting, "A comparative study of message digest 5 (md5) and sha256 algorithm," in *Journal of Physics: Conference Series*vol. 978, IOP Publishing, Article ID 012116, 2018.

[30] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[31] W.-M. Lee, "Using the web3. js apis," in *Beginning Ethereum Smart Contracts Programming*, pp. 169–198, Springer, Berlin, Germany, 2019.

[32] O. Hasan, L. Brunie, and E. Bertino, "Privacy Preserving Reputation Systems Based on Blockchain and Other Cryptographic Building Blocks: A Survey," University of Lyon, INSA-Lyon, CNRS, LIRIS, UMR5205, Lyon, France, 03034994, 2020.

[33] D. J. Bernstein, C. Chuengsatiansup, and T. Lange, "Curve41417: Karatsuba revisited," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 316–334, Springer, Berlin, Germany, 2014.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7001278, 12 pages
https://doi.org/10.1155/2022/7001278

Research Article

Noise-Resistant Video Channel Identification

Mingkai Wang,[1] **Zengkun Xie**,[2] **Xiangdong Tang**,[1] **and Fei Chen**[1]

[1]*College of Computer Science & Technology, Qingdao University, Qingdao, China*
[2]*Department of electrical and new energy engineering, Yantai Engineering &Technology College, Yantai, China*

Correspondence should be addressed to Fei Chen; feic@qdu.edu.cn

Received 30 January 2022; Accepted 22 June 2022; Published 18 August 2022

Academic Editor: Zengpeng Li

As the video streaming traffic grows exponentially nowadays, variable bitrate (VBR) encoding has been widely utilized by modern live video streaming service providers, such as YouTube, TikTok, and Twitch. However, video bitrate can be a delicate fingerprint of the video streaming, leading to risks of privacy leakage. There are several studies that attempt to eavesdrop the privacy from encrypted video streaming, but most of them presume strict requirements on the implementation environments and have great limitations when noise interference exists. Actually, the video traffic from the multimedia edge server is distinct from inter-application traffic flows due to device customization and can be identified even if there are noise interferences or the victim in a weak network condition. In this paper, a video traffic identification method is proposed to identify the encrypted video streaming from multimedia edge server under the interference of irrelevant traffic flows. Initially, we use an interapplication filter to identify the traffic from the edge server. Then, a longest-common-subsequence (LCS)-based method is developed for similarity matching to resist the noise interference from unpredictable burst traffic and network environment variations. In order to evaluate the system performance, we setup the prototype system with an AWS EC2 server and a raspberry pi device, then utilize the real-world trace data for pushing movies to victims. The experimental results show that the accuracy of our proposed strategy can reach 89.1% within 140 seconds eavesdropping even mixed with 14% noise interference.

1. Introduction

With the improvement of the network bandwidth, the video streaming service has been popular in recent years, which quickly sweeps across the world and takes up the viewers' free time by high-quality content in live e-commerce, sports events, or video games. For example, according to the report of Statista, which is a global business data platform, shows that the number of monthly active users of TikTok worldwide has exceeded 1 billion [1]. Meanwhile, the number of monthly active users of YouTube has exceeded 2.3 billion. However, the growing number of users has brought great bandwidth pressure to video data center. Thanks to the development of edge computing in recent years, more and more Internet service providers try to save server resources and reduce the round-trip time by handing user tasks to edge servers, such as computation offloading [2] and video delivery [3]. In the foreseeable future, more and more applications will be handled by edge servers with the performance improvement of edge devices and popularity of 5G infrastructure.

Conventionally, the bitrate-based fingerprint carried by video traffic flow can be identified by video traffic pattern analysis even with the transportation layer security (e.g., TLS) encryption. There are many studies attempting to eavesdrop the content of videos from viewers which are under TLS encryption in recent years [4–6], but most of these works assume that the encrypted video stream can be directly observed by attackers without interference of irrelevant traffic flows. Some studies also proposed noise-resistant fingerprint identification methods, but all of them are not suitable for video bitrate fingerprints [7]. Actually, the video traffic is usually delivered from content delivery network (CDN) which may serve multiple websites or applications at the same time [8]. Therefore, the complete and noiseless bitrate-based traffic fingerprint can be hardly identified from the real-world trace data. Furthermore, the effectiveness of traffic fingerprints is highly sensitive to network fluctuations, and the partial features of traffic fingerprint will drift seriously during unstable network conditions.

The prevalence of edge server brings a new risk to video traffic identification due to the customization of the edge devices. Conventionally, the CDN server usually undertakes several tasks including video delivery and static resource delivery using the same domain name, which will make it difficult to identify the video traffic flow encrypted by TLS. However, the multimedia edge server hardly delivers the irrelevant traffic due to the customization of the edge device, which leads to the possibility of identifying the bitrate-based traffic flows from it. Therefore, the video traffic from the edge server is easier to identify and the traffic features are more stable. In this paper, we will present a noise-resistant video traffic identification method for VBR traffic flow. We will show that the traffic fingerprint from the real-world trace data captured from multimedia edge server can also match the bitrate fingerprint after appropriate preprocess. Initially, a simple traffic filter which only uses three labels from the unencrypted traffic is used to filter out the traffic that is from the multimedia edge server. After that, an LCS-based fingerprint-matching method is proposed to eliminate the interference of the remaining two types of noise and match the traffic fingerprint and bitrate fingerprint.

The rest of this paper is organized as follows: The literature is explored in Section 2. The data analysis is presented in Section 3. The system design is presented in Section 4. The traffic filter and LCS-based matching method are illustrated in Section 5 and Section 6. The system performance is evaluated in Section 7. Finally, Section 8 concludes this paper.

2. Related Work

2.1. Privacy Leakage and Protection. With the growth of Internet applications, new security issues arise with the development of Internet infrastructures. On the one hand, the new paradigms could bring facilities to our daily life such as recommendation system [9, 10], computation offloading [2, 11, 12], and route planning [13, 14]. On the other hand, the privacy defense strategy also needs to consider more aspects with the upgrading of infrastructure: mobile devices [15], Internet of things (IoT) device [16–18], and cloud server [14, 19]. Specifically, machine learning [20] and edge computing are developed rapidly, which brings more complex privacy leakage problems [21]. With the improvement of bandwidth and device performance, more video streaming service providers use edge servers to cache and distribute video content in order to reduce the pressure of data center, which leads to the popularity of research of multimedia privacy protection on edge server [22, 23]. In this paper, we will discuss the privacy leakage caused by encrypted video under noise interference.

2.2. Privacy Leakage from Video Stream. The side channel attack caused by privacy leakage of encrypted video has attracted extensive attention in recent years. Saponas et al. [4] makes fingerprints by using multiple sliding windows to divide the video into segments of several milliseconds based on VBR encoded video, but they only achieve 62% accuracy with 10 minutes eavesdropping without noise interference. Gu et al. [24] improved the DTW algorithm to make it suitable for DASH protocol and made a classifier that can identify videos from both Netflix and YouTube, but they claim that the low bandwidth and high packet loss rate are not in their consideration since users will normally leave video streaming immediately because of the bad experience. As the prevalence of machine learning, neural network has an advantage of feature extraction in a sophisticated environment. Schuster et al. [5] modeled the fingerprints and proposed a CNN-based model to identify the fingerprints for VBR-based videos from YouTube and Netflix. Nevertheless, all the bitrate-based video identification strategies need the assumption of stable network. Otherwise, both weak network condition and burst traffic will have a serious interference on traffic fingerprint, which will inevitably lead to wrong identification results because the points in bitrate fingerprint will be matched incorrectly. In the following part, we will analyze the noise interference and then propose a noise-resistant video traffic identification method.

2.3. Sequence Matching Method. Sequence matching methods are essential in solving many pattern recognition problems such as anomaly detection, speech recognition, and other domains [25]. The popular methods usually consider using points for matching (e.g., Edit Distance on Real Sequence (EDR) [26], Dynamic Time Warping (DTW) [27]), using shape for matching (e.g., Frechet distance [28]), and segmenting the sequence for matching (e.g., One Way Distance [29]). Nevertheless, most sequence matching methods do not consider the matching effectiveness in interference environment. Thanks to the powerful representation ability of deep learning, similarity learning can accommodate heterogeneous features in the sophisticated environments, and there are several deep-learning-based methods like the CNN-based solution [30, 31], and the LSTM-based solution [32]. However, deep-learning-based models usually need online training to adapt the latest features, and the computational cost is very high.

3. Traffic Data Analysis

In this section, we will introduce the video data analysis to illustrate the video bitrate and several types of traffic noise using the classic movie *Titanic*. In the following parts, *nginx* and *ffmpeg* is used to push the encrypted video traffic, *Chrome downloader* is used to provide the irrelevant traffic, and *wondershaper* is used to simulate the weak network environment with the random interference of bandwidth limitation, RTT, and packet loss.

3.1. Side Channel Attack on Video Traffic. VBR can bring the risk of privacy leakage through the bitrate fluctuation. Figure 1 shows the bitrate of a video which encode with constant bitrate (CBR) and VBR, and it can be seen that there are significantly different fluctuation trends between them.

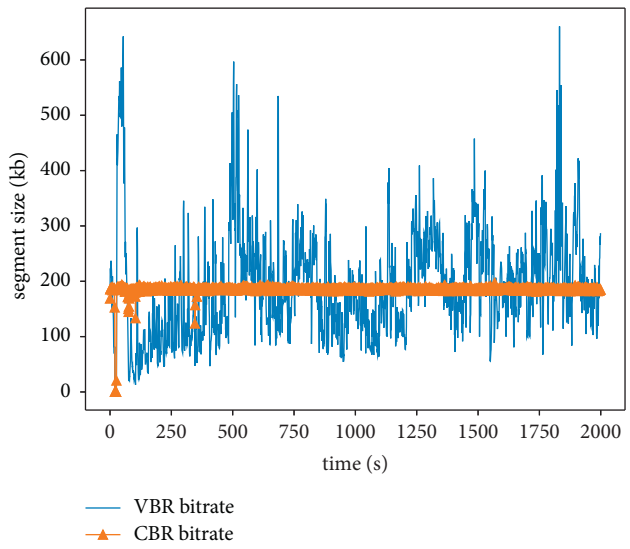


Figure 1: A comparison of CBR and VBR bitrate.

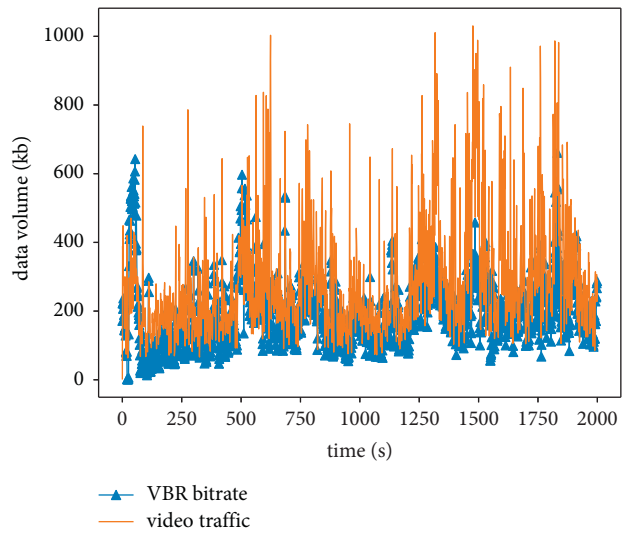


Figure 2: Video bitrate and traffic.

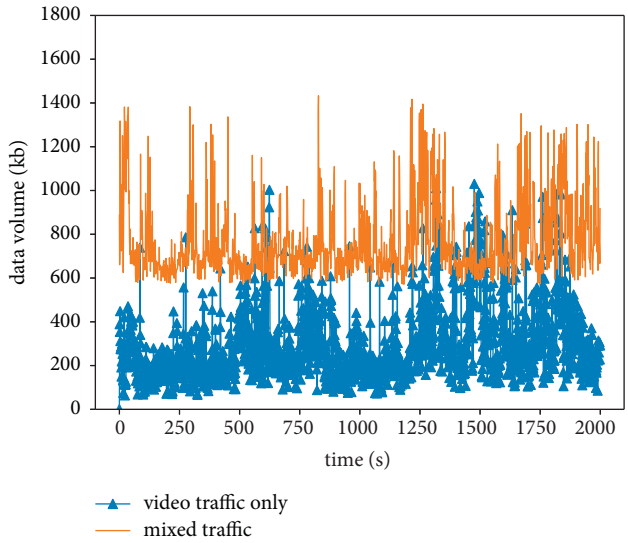


Figure 3: Video traffic and mixed traffic.

Additionally, TLS only encrypts the content, but leak the statistical features of the traffic. Figure 2 shows the correlation between the bitrate of VBR video and it is encrypted video streaming.

Obviously, the privacy of video viewers can be identified through the analysis of the video traffic even after encryption. When the attacker obtains a traffic fingerprint segment, the privacy may be leaked.

3.2. Bitrate Features with Irrelevant Traffic. When providing video streaming services for users, edge devices can also provide other multimedia services from different websites at the same time (such as encode offloading or download acceleration), resulting in the eavesdropped traffic containing multiple types of packets, which make it difficult to identify the video traffic. Figure 3 shows the traffic from a raspberry edge server, which contains only video stream and both video stream and download stream.

It can be seen that the video stream traffic is covered by mixed traffic, resulting in the disappearance of the video traffic features.

3.3. Bitrate Features in the Weak Network Condition. VBR features are usually easy to identify, which is more likely to lead to privacy leakage. However, such features are easily affected by noise or weak network condition, which reduces the accuracy of identification. Figure 4 shows the interference of bandwidth limitation and RTT on the traffic fingerprint of *Pirates of the Caribbean 5* from 1000 seconds to 1700 seconds. The video traffic is collected from raspberry edge server.

Since the beginning of traffic eavesdropping, the bandwidth limitation from 50 to 120 second and the burst RTT from 170 to 180 second lead to video playback jitter and corresponding backward drift of traffic features. Figure 5 adds the interference of 15% random packet loss to the traffic fingerprint of *Pirates of the Caribbean 5* from 2000 seconds to 2200 seconds. Due to the packet retransmission function of TCP protocol, the interference of feature drift is reduced, but it still reduces the matching accuracy between bitrate fingerprint and traffic fingerprint. In a word, the bitrate-based video fingerprints raise stringent requirements on network conditions.

3.4. Bitrate Features with Intra-Application Interference. Even in the same application, the features will also be significantly affected by user operations, which usually cannot be predicted. Whether viewers explore the video list while watching or communicating through the intrasite chat system, it will have a destructive interference on the traffic features and seriously reduce the identification accuracy. Figure 6 shows the burst traffic by browsing the video list and the interference on the traffic features.

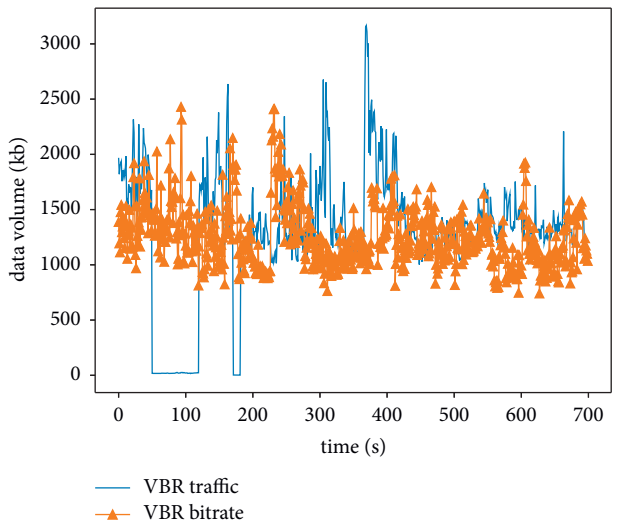


Figure 4: Traffic features under bandwidth limitation and RTT.

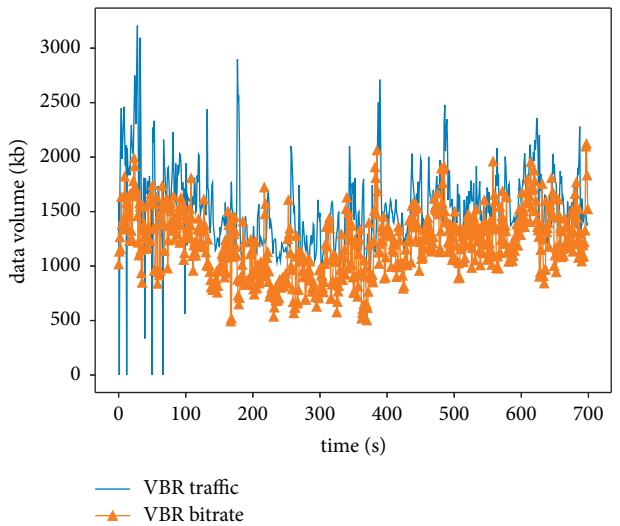


Figure 5: Traffic features under packet loss.

Obviously, the traffic generated by unpredictable behavior on 100 s to 110 s completely covers the original traffic features.

4. System Design

In this section, we will present the system design with the noise-resistant encrypted video traffic identification. The system structure is presented in Figure 7. The proposed system can be divided into following parts:

(i) Interapplication traffic filter: A filter based on three labels including server name indicator (SNI) is proposed to filter out the traffic that from the multimedia edge server.

(ii) LCS-based fingerprint matching: An LCS-based method is proposed for matching the traffic fingerprint and bitrate fingerprint under noise interferences.

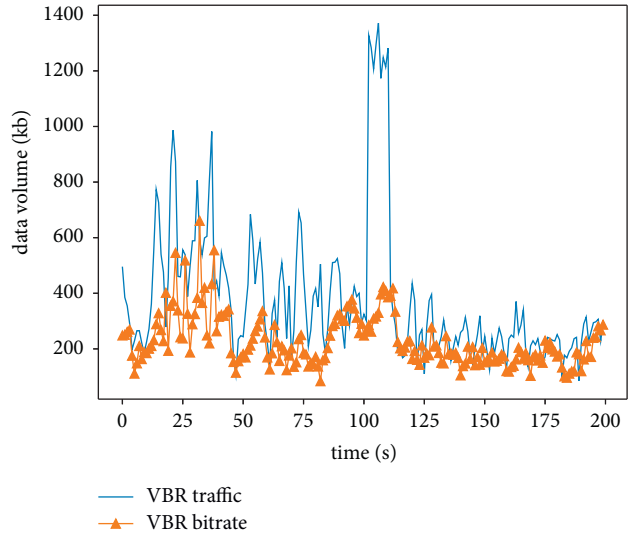


Figure 6: Burst traffic caused by user behavior.

The SNI tag is used to bring the domain name requested by the server through a plain text in the handshake stage of the TLS protocol. The attacker can easily obtain the target domain using SNI as an interapplication traffic filter, and further identify the whole TLS session through IP address or sequence number, and then obtain the video traffic flow completely without other interinterference due to the customization of the edge device. It should be noted that all the video providers need to transfer the video stream according to the protocol which specified by the edge multimedia framework, and the edge server will use the unified video protocol to send the video stream to users. As the popular edge multimedia frameworks such as *EasyNVR* or *Link Visual* all use TLS for video delivery, so our filter can be regarded as a general method for the existing video service. However, the traffic fingerprint will still affect by the burst traffic from unpredictable behaviors (such as exploring the video list), or the weak network condition, for example, low bandwidth and packet loss after filtering. So an LCS-based method is proposed to filter the intraapplication interference and identify the matched segments between traffic fingerprint and bitrate fingerprint.

5. Interapplication Traffic Filter

We will propose a traffic filter to eliminate irrelevant traffic from other applications in this section. Three labels are utilized to achieve the traffic filter: SNI, content type, and source IP address (srcaddr):

(i) SNI is used to filter the video traffic which to be identified.

(ii) ContentType is used to divide the TLS session.

(iii) IP address is used to obtain the continuous TLS session.

The video content is sent in stream, but each video segment is encrypted in a TLS session, thus, the session is denoted as a video segment in a fixed length. ContenType is

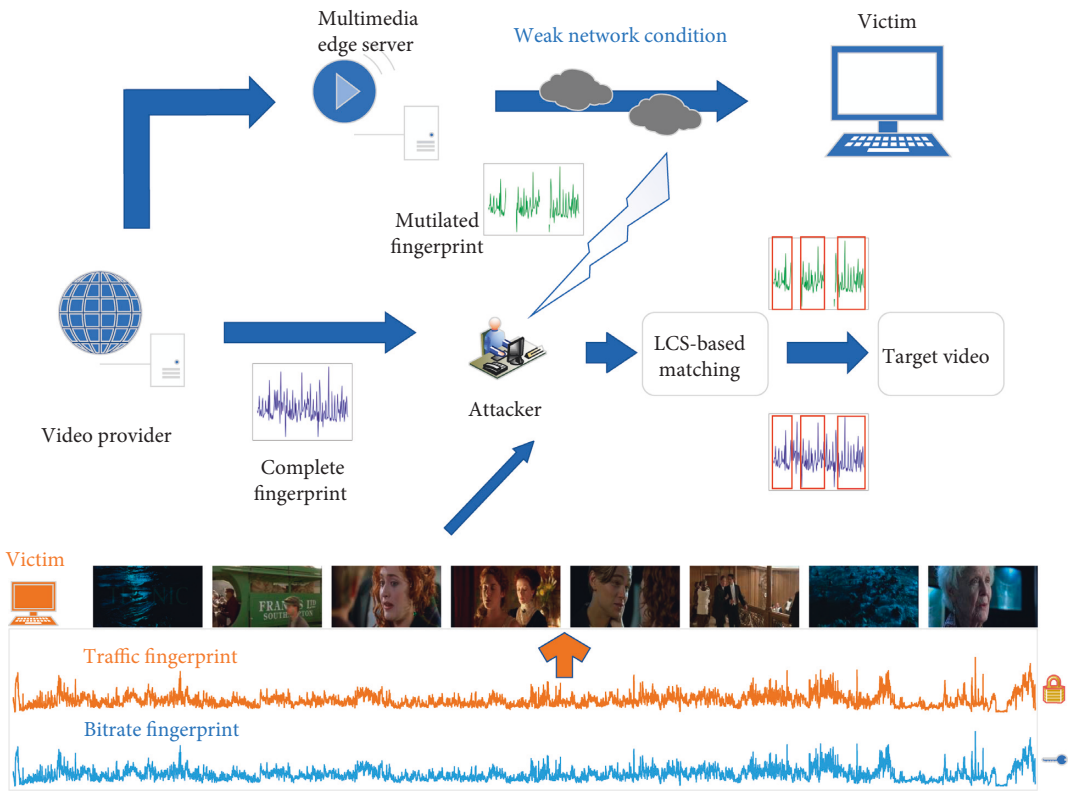


Figure 7: An overview of system structure.

used to check whether the packet is a TLS handshake packet (denote the start of a new TLS session). Since the SNI in the handshake packet holds the source domain name without encrypted, all video streaming TLS sessions can be identified. The filtering process is shown in Algorithm 1.

After filtering, we get a set $S = \{s_0, s_1 \ldots s_j\}$ containing j packets in all TLS sections, where s_j is a two-tuple $\langle \text{length}_j, \text{time}_j \rangle$ for the j th packet with t_j as the arrival time and $length_j$ as the packet length.

6. Noise-Resistant Fingerprint Matching

In the previous section, we obtain the packet sequence through filtering the TLS session. However, the intra-application interference still exists and seriously reduces the matching accuracy. In this section, we will propose a noise-resistant similarity matching method based on LCS model. Before performing the matching model between bitrate fingerprint and traffic fingerprint, we should discuss the feature drifting caused by weak network condition and intraapplication noise interference. The bandwidth fluctuation caused by weak network will limit the data obtained by viewers and then destroy the traffic fingerprint. For example, for the same video segment which bitrate fingerprint is (1, 2, 3, 4, 5), the traffic fingerprint eavesdropped from a viewer with stable network is (2, 3, 4, 5, 6), but eavesdropped from another viewer with weak network will become (2, 3, 0, 0, 4, 5, 6), which will seriously reduce the identification accuracy. Similarly, the intraapplication noise will also change the traffic features and reduce the accuracy. For example, the traffic fingerprint eavesdropped from a viewer without interference is (2, 3, 4, 5, 6), but when there is a burst traffic caused by unpredictable behavior, the traffic fingerprint will cover by burst traffic interference and become (2, 7, 11, 8, 6). The two types of interference above refer to the drift between bitrate fingerprint and traffic fingerprint which violates the uniqueness in a fine granularity observation, even though the trend keeps consistent in the long-term observation. In order to perform the similarity matching method, we relocate the traffic fingerprint by second, as shown in Algorithm 2.

The algorithm recalculates the length of the packet in sequence S and matches the element in bitrate fingerprint with the timeline. Generally, weak networks and burst traffic are infrequent, it means that if most intervals of traffic fingerprint and bitrate fingerprint are matched in the long-term trend, we can ignore a few local mismatch caused by weak network or burst traffic. However, the common similarity matching method requires that all the elements in the sequence must be matched even if the fingerprint is under interference. Therefore, we propose a fingerprint matching method considering the traffic noise interference. We define $F(x_a, x_b)$ as the Euclidean distance between x_a and x_b. For a given x_a and x_b, if $F(x_a, x_b)$ is less than threshold ϵ, the x_a is considered to match x_b. Then, a noise-resistant model *N-LCS* based on LCS model is proposed to adapt the fingerprint mismatches.

First, for the bitrate fingerprint T^B and traffic fingerprint T^F, the points in T^B can only match the points in T^F forward (e.g., T^B_5 can only match $T^F_{5,6,7\ldots}$). This is because during the

video playback, the video player will not cache the played video contents. In addition, the matching strategy of LCS is too simple to adapt the weak network condition, so N-LCS optimize the matching strategy to adapt the noise interference. For $T^B = \{t_0^b, t_1^b, \ldots, t_d^b \ldots\}$ and $T^F = \{t_0^f, t_1^f, \ldots t_c^f \ldots\}$:

(i) if $t_c^f > t_d^b$ and $F(t_c^f, t_d^b) < \epsilon$, the point t_c^f and t_d^b are considered to be matched.

(ii) if $t_c^f > t_d^b$ and $F(t_c^f, t_d^b) < \epsilon$, the point t_c^f and t_d^b are considered to be not matched, and the unmatched point t_d^b may have been caused by burst traffic.

(iii) if $t_c^f < t_d^b$ and $F(t_c^f, t_d^b) < \epsilon$, the point t_c^f and t_d^b are considered to be not matched, and the unmatched point t_d^b may caused by limited bandwidth, RTT or packet loss. As the limited traffic will usually lead to the drift of traffic features, and the backtracking function should be added to LCS model in order to drop the redundant fingerprint at the trail of T^B to avoid false matching.

We use a two-dimensional matrix M with the size of $k * k$ to save the temporary matching result, where k is the length of bitrate and traffic fingerprint. The values of matrix M are calculated by the following formula:

$$M[i][j] = \begin{cases} M[i-1][j-1], \\ F\left(t_i^f, t_j^b\right) > \epsilon \text{ and } t_i^f > t_j^b, \\ M[i-1][j-1] + 1, \\ F\left(t_i^f, t_j^b\right) < \epsilon \text{ and } t_i^f > t_j^b, \\ \max\left(M[i1][j-1], M[i-1][j]\right), \\ F\left(t_i^f, t_j^b\right) < \epsilon \text{ and } t_i^f < t_j^b, \\ 0, \\ i = 0 \text{ or } j = 0, \end{cases} \quad (1)$$

$$0 < i,\ j <= k,$$

where ϵ is the threshold of F. In order to eliminate the interference of feature drift, N-LCS makes two rounds of backtracking at the end of the algorithm. The first round of backtracking determines the drift distance of the traffic fingerprint and drops the fingerprint at the tail of the bitrate fingerprint according to the drift distance. The second round of backtracking will use the bitrate fingerprint calculated in the first round to find the matching path in the matrix M and calculate the longest common subsequence between two fingerprints according to the new matching path using dynamic programing as the matching result. The calculating process is shown in Algorithm 3.

Figure 8 shows the partial match result between traffic fingerprint and bitrate fingerprint. The red line shows the match relation between bitrate and traffic fingerprint. It can be seen that the LCS-based matching model can successfully ignore the invalid features caused by interference.

7. Implementation and Evaluation

7.1. Experimental Setup. In order to build the prototype system, we have an Amazon EC2 server as the video stream server, a raspberry pi as the edge server, and an Xiaomi 11 Ultra as the victim, respectively. The server configuration is listed in Table 1. *nginx* and *ffmpeg* is used to push the video streaming in RTMPS protocol, and *Wireshark* is used to simulate Man-In-The Middle (MITM) attack to capture the encrypted TLS traffic of the victim. We use videos with several bitrates to generate the bitrate and traffic fingerprint and evaluate the effectiveness of N-LCS, and the configuration of video dataset is shown in Table 2 (The data set can be found at https://1drv.ms/u/s!AnB84OgJQM04jkAYDlzO9fhchxeZ?e=fj4cY8).

7.2. Effectiveness of the Traffic Filter. Then we test the effectiveness of the interapplication traffic filter proposed in Section 5. We use *Wireshark* to capture the video traffic encrypted by RTMPS protocol, and A domain name registered from Tencent cloud is used to fill in the SNI tags. The output traffic from the edge device and the filtered input traffic from the victim are collected, respectively, as shown in Figure 9. The results show that the proposed traffic filter can identify all the target TLS sessions accurately.

7.3. Threshold Analysis. In this part, we will calculate the threshold ϵ of N-LCS model, which is used to identify the matched point in traffic fingerprint and bitrate fingerprint. A total of 300 groups of 50 seconds bitrate fingerprints and traffic fingerprints are used to calculate the similarity distance in the following cases, and the similarity distance is shown in Figure 10:

(i) Fingerprints come from the same video.

(ii) Fingerprints come from different videos, but the bitrate is similar.

(iii) Fingerprints come from different videos, and the bitrate of different videos varies greatly.

Then, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) is used to define accuracy:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (2)$$

After that, we use the intersection of two false rate lines as the threshold to maximize the accuracy. As shown in Figure 11, 239 is the best threshold to reach the maximum accuracy of 0.766 (76.6% points in the fingerprints can be accurately matched).

```
Input:
  packet sequence P;
Output:
  packet sequence S;
(1)  while packet [++i] != NULL do
(2)    if ContentType == HandShake and SNI == target domain then
(3)      Create a new sequence s
(4)      Old_IP = packet [i]. ip
(5)    else if ContentType != HandShake and packet [i].ip == Old_IP then
(6)      Add packet [i]. length to sequence
(7)    end if
(8)  end while
```

Algorithm 1: Video filter.

```
Input:
  packet sequence S;
Output:
  traffic fingerprint T^f;
(1)  old_time = 0
(2)  acc_len = 0
(3)  while packet [++i] != NULL do
(4)    ifpacket [i].time - old_time ≥ 1 then
(5)      T^f. append(acc_time)
(6)      acc_len = 0
(7)      old_time ++
(8)    else if packet [i].time - old_time < 1 then
(9)      acc_len += packet [i]. length
(10)   end if
(11) end while
```

Algorithm 2: Traffic fingerprint relocator.

Finally, we calculate the identification accuracy with 1–100 matching points as the threshold δ in above three cases, and the results are shown in Figure 12. When the bitrate of different videos varies greatly, there are less matched points between fingerprints, and the similarity distance between mismatched points is usually large, so only a small threshold is required to achieve high accuracy. When the bitrate is similar and the length of fingerprints is short, there are also many matched points though the fingerprints that come from different videos, result in the a lower accuracy compared with other cases. Since the identification accuracy of the threshold for matching points is not 100%, the identification accuracy will eventually decrease to 0 with the increase of threshold δ. Considering the difference between fingerprints, we use 0.43 as the threshold δ in following experiments.

7.4. The Effectiveness of N-LCS without Noise Interference. Figure 13 compares the N-LCS with two popular similarity-matching methods in a noise-free environment with different fingerprint lengths.

With the increase of fingerprint length, the proportion of matched segments in fingerprints gradually stabilizes, so the accuracy of all algorithms are increasing. However, the focus of N-LCS is to identify and remove the noise interference in the traffic fingerprint, rather than improve the matching accuracy of fingerprints without noise interference; therefore, the accuracy of N-LCS is close to Pearson. It is worth noting that the fluctuation of traffic features lead to the poor performance of DTW algorithm based on global optimal distance, and the accuracy is significantly lower than Pearson and N-LCS.

7.5. The Effectiveness of N-LCS under Noise Interference. In order to evaluate the effectiveness of N-LCS under the noise interference, we use the automatic script to randomly generate different levels of noise interference during video playback. The fingerprint with a length of 200 seconds is used to test the interference of bandwidth limitation, burst RTT, packet loss, and burst traffic on the identification accuracy of N-LCS under different noise levels. The results are shown in Table 3 then, the traffic captured with mixed noise (bandwidth limitation, packet loss and burst traffic account for 1/3 respectively) is used to compare the N-LCS, Pearson, and DTW algorithms. The results are shown in Figure 14.

With the increase in the proportion of noise interference, the identification accuracy of above algorithms

Input:
bitrate fingerprint T^B;
traffic fingerprint T^F;
Output:
the length of subsequence Result
(1) k = len (T^B); define matrix M [k][k] and pre [k][k]
(2) **for** iterate T^F and T^B **do**
(3) **if** T^F [i] - T^B [j] < ϵ **then**
(4) **if** T^F [i] > T^B [j] **then**
(5) M [i][j] = M [$i-1$][$j-1$] + 1, mark i and j as matched points in matrix pre
(6) **else**
(7) M [i][j] = M [$i-1$][$j-1$]
(8) **end if**
(9) **else if** M [$i-1$][j] > M [i][$j-1$] **then**
(10) M [i][j] = M [$i-1$][j]
(11) **else**
(12) M [i][j] = M [i][$j-1$], mark i and j as noise points in matrix pre
(13) **end if**
(14) **end for**
(15) $i = T^F$.length; $j = T^B$.length
(16) **while** iterate similarity path in pre **do**
(17) **if** pre [i][j] holds noise points **then**
(18) tmp ++
(19) **end if**
(20) **end while**
(21) $i = T^F$.length - tmp; $j = T^B$.length
(22) **while** iterate similarity path in pre **do**
(23) **if** pre [i][j] holds matched points **then**
(24) Result ++
(25) **end if**
(26) **end while**

Algorithm 3: N-LCS solver.

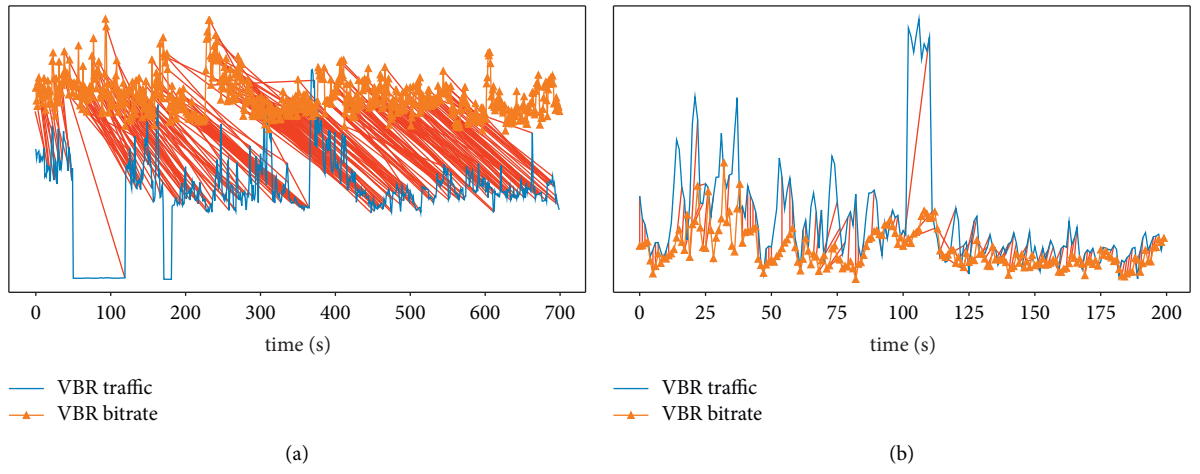


Figure 8: The comparison of matching result. (a) Video traffic in weak network conditions. (b) Video traffic with burst noise interference.

decreased in varying degrees, while the accuracy of DTW and Pearson decreased much faster than N-LCS. In addition, when the proportion of noise interference is less than 14%, the accuracy of N-LCS decreases slowly, while when the proportion exceeds 15%, the accuracy decreases significantly. This is because the N-LCS matching strategy reserves sufficient redundant for noise interference. The average number of matching points between matched fingerprints is much higher than the identification threshold, and it will not have a great interference to the accuracy though there is a small amount of unmatched points. Then, we set the noise proportion to 14%, and compare N-LCS with three latest identification methods based on video fingerprint: beauty [5], p-dtw [24], and

Table 1: Sever configuration.

	ec2	raspberry pi
System	Windows server 2019	Ubuntu 18.04
Memory	1 GB	1 GB
Cpu	2.5 GHZ * 1	1.2 GHZ * 4
Hard disk	30 GB	16 GB
Network bandwidth	10 mbps	100 mbps

Table 2: Video dataset.

	Time	bitrate
Pirates of the Caribbean 5	02:48:30	10.1 mbps
Pirates of the Caribbean 5	02:48:30	8005 kbps
Pirates of the Caribbean 5	02:48:30	5991 kbps
Pirates of the Caribbean 5	02:48:30	4022 kbps
Pirates of the Caribbean 5	02:48:30	2074 kbps
Titanic	03:06:49	2607 kbps
Inception	02:28:21	1986 kbps
Avengers 3	02:29:33	2217 kbps
Trainspotting	01:34:16	1825 kbps

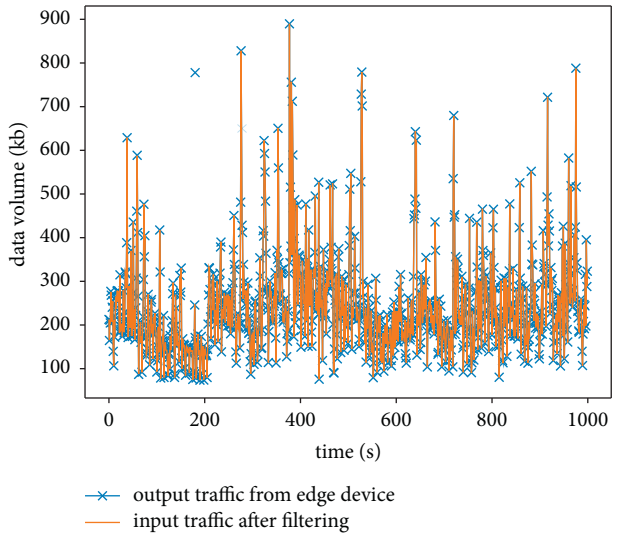


Figure 9: The comparison of output traffic from the edge devices and filtered traffic from the victim.

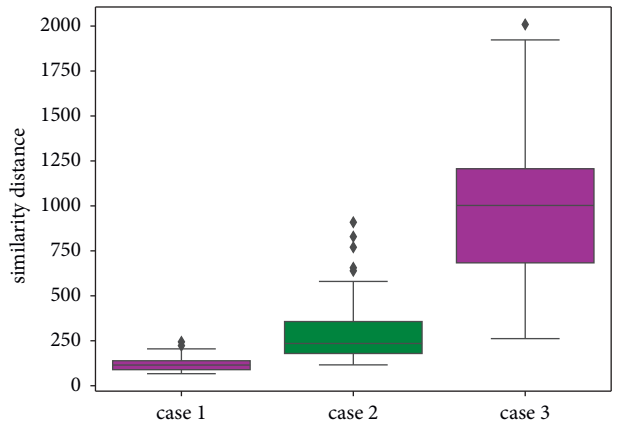


Figure 10: Similarity distance of our method.

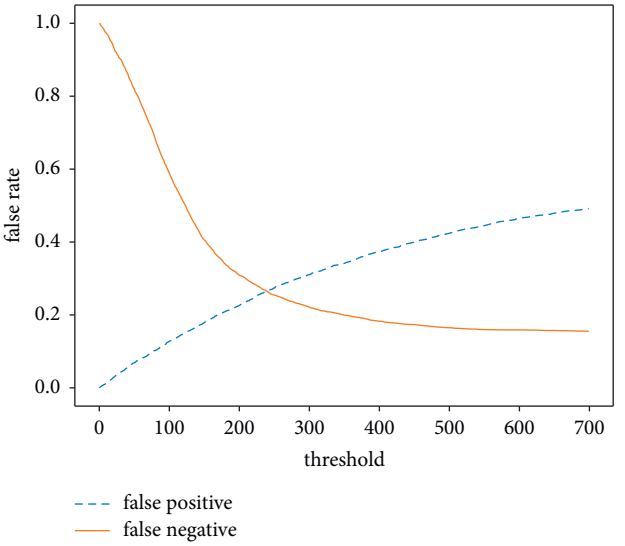


Figure 11: False rate of N-LCS.

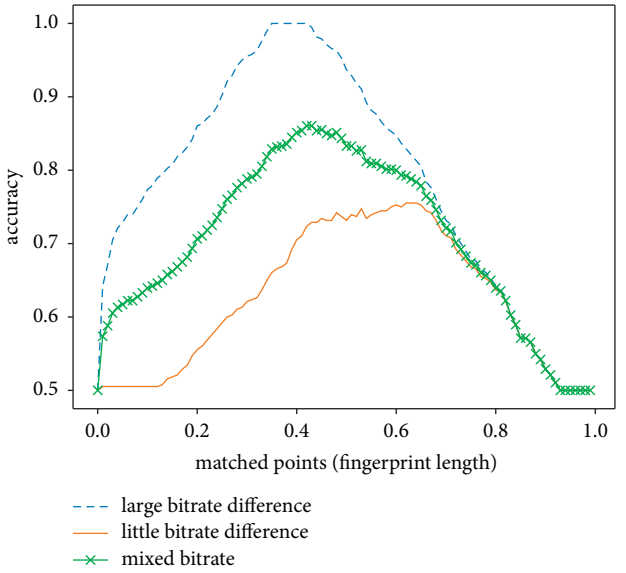


Figure 12: Accuracy in different bitrates.

leaky [33]. The test video clips were taken from the films *Titanic*, *Pirates of the Caribbean 5*, *Inception* and *Avengers 3*. The results are shown in Table 4. As the previous methods only focus on the accuracy of matching strategy, ignoring the noise interference from the real-world eavesdropping environments, result in the

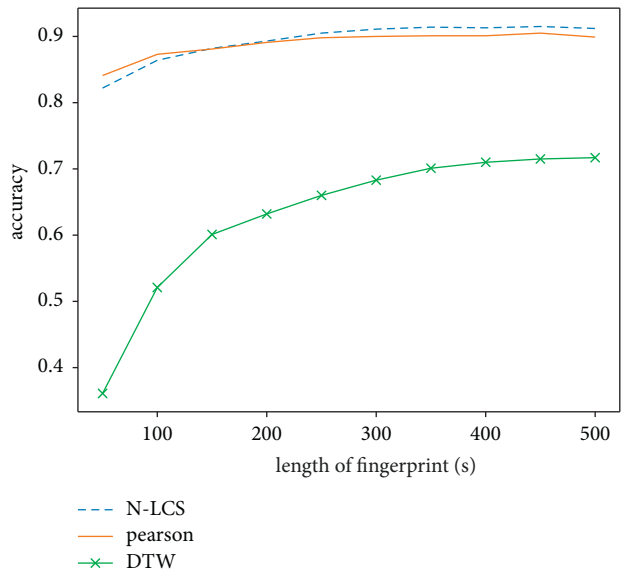


Figure 13: Accuracy of different methods without noise interference.

Table 3: Accuracy of N-LCS under different noise levels.

	2%	4%	6%	8%	10%	12%	14%	16%	18%	20%
Bandwidth limitation	0.870	0.861	0.859	0.855	0.845	0.837	0.821	0.781	0.733	0.679
Burst RTT	0.868	0.870	0.856	0.845	0.839	0.833	0.821	0.779	0.724	0.661
Packet loss	0.909	0.904	0.905	0.876	0.881	0.874	0.861	0.830	0.806	0.778
Burst traffic	0.883	0.877	0.850	0.849	0.840	0.827	0.815	0.767	0.718	0.650

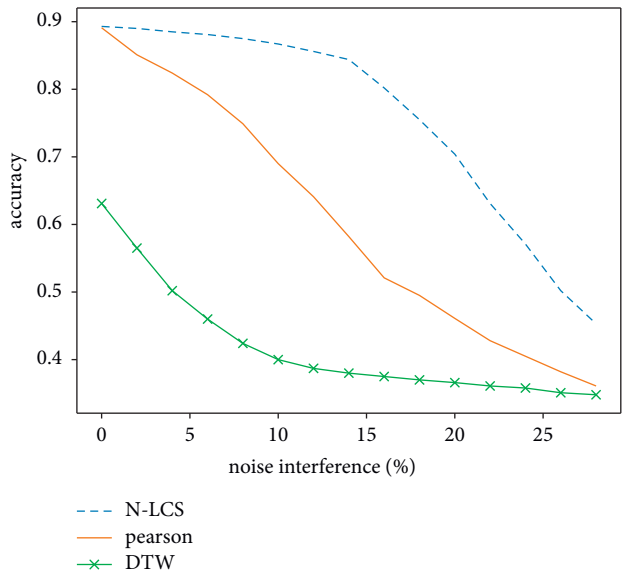


Figure 14: Accuracy of different methods under noise interference.

Table 4: Accuracy of different methods under noise interference.

	50 s	60 s	70 s	80 s	90 s	100 s	110 s	120 s	130 s	140 s
N-LCS	**0.649**	**0.711**	**0.776**	**0.821**	**0.850**	**0.872**	**0.883**	**0.887**	**0.889**	**0.891**
Beauty	0.369	0.461	0.545	0.618	0.682	0.717	0.751	0.772	0.791	0.807
P-DTW	0.420	0.491	0.553	0.605	0.649	0.677	0.701	0.721	0.737	0.752
Leaky	0.349	0.398	0.452	0.483	0.510	0.531	0.545	0.558	0.564	0.562

Here, the data unit is percentage. So, 0.649 means the accuracy is 64.9%. The bolded values represent the highest value for each column.

reduction of accuracy in the weak network condition and N-LCS can reach the highest accuracy even under noise interference.

8. Conclusion and Future Work

In this paper, we proposed a noise-resistant bitrate-based identification method for encrypted video traffic on the raspberry pi platform, which uses the LCS-based model to match the traffic and bitrate fingerprint. A real dataset using several famous movies captured from edge server and a prototype system was presented for performance evaluation. Through experiments, we proved that even the interference proportion can reach to 14%, and we can reach 89.1% accuracy after 140 seconds traffic eavesdropping.

With the prevalence of video streaming system, our work provides a new eavesdropping method that robust to interference. In the future work, we will optimize our model from the following two aspects. First, the identification accuracy will be optimized when the traffic fingerprints eavesdropped from victims are similar. Second, the proposed method only supports the popular protocols used in multimedia edge frameworks such, as RTMPS, and more protocols will be supported, for example, HLS and DASH in the future.

Data Availability

The bitrate fingerprint data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Zengkun Xie performed data collection, cleaning, and annotation, which are important parts of our work and greatly support our proposed machine learning methods.

Acknowledgments

This work is supported by National Natural Science Foundation of China 61602214.

References

[1] "Statistic Social Media & User-Generated Content," 2021, https://www.statista.com/statistics/1267892/tiktok-global-mau/.

[2] P. Becvar and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, pp. 1628–16563, Berlin, Germany, October 2017.

[3] Y. Li, P. A. Frangoudis, Y. Hadjadj-Aoul, and P. Bertin, "A mobile edge computing-based architecture for improved adaptive http video delivery," in *Proceedings of the 2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, IEEE, 2016.

[4] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno, "Devices that tell on you: privacy trends in consumer ubiquitous computing," in *Proceedings of the 16th USENIX Security Symposium on USENIX Security Symposium, SS'07*, USENIX Association, Boston, MA, USA, August 2007.

[5] R. Schuster, V. Shmatikov, and E. Tromer, "Beauty and the burst: remote identification of encrypted video streams," in *Proceedings of the 26th USENIX Security Symposium (USENIX Security 17)*, pp. 1357–1374, Vancouver, Canada, 2017.

[6] H. Wu, Z. Yu, G. Cheng, and S. Guo, "Identification of encrypted video streaming based on differential fingerprints," in *Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, July 2020.

[7] Q. Lu, S. Li, J. Zhang, and R. Jiang, "Pedr: exploiting phase error drift range to detect full-model rogue access point attacks," *Computers & Security*, vol. 114, Article ID 102581, 2022.

[8] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, "Dissecting video server selection strategies in the youtube cdn," in *Proceedings of the 2011 31st International Conference on Distributed Computing Systems*, IEEE, Minneapolis, MN, USA, June 2011.

[9] J. Davidson, B. Liebald, J. Liu et al., "The youtube video recommendation system," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, Barcelona, Spain, September 2010.

[10] D. Das, L. Sahoo, and S. Datta, "A survey on recommendation system," *International Journal of Computer Application*, vol. 160, no. 7, pp. 6–10, 2017.

[11] Y. Zheng, C. Tian, H. Zhang, J. Yu, and F. Li, "Lattice-based weak-key analysis on single-server outsourcing protocols of modular exponentiations and basic countermeasures," *Journal of Computer and System Sciences*, vol. 121, pp. 18–33, 2021.

[12] C. Tian, J. Yu, H. Zhang, H. Xue, C. Wang, and K. Ren, "Novel secure outsourcing of modular inversion for arbitrary and variable modulus," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 241–253, 2022.

[13] D. Zhang, C.-Y. Chow, Q. Li, X. Zhang, and Y. Xu, "A spatial mashup service for efficient evaluation of concurrent k-nn queries," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2428–2442, 2015.

[14] X. Cheng and X. Cheng, "A secure and lightweight data sharing scheme for internet of medical things," *IEEE Access*, vol. 8, pp. 5022–5030, 2020.

[15] X. Lu, Z. Pan, and H. Xian, "An integrity verification scheme of cloud storage for internet-of-things mobile terminal devices," *Computers & Security*, vol. 92, Article ID 101686, 2020.

[16] Z. Wang and D. Wang, "Achieving one-round password-based authenticated key exchange over lattices," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 308–321, 2022.

[17] Z. Li, M. Alazab, S. Garg, and M. S. Hossain, "Priparkrec: privacy-preserving decentralized parking recommendation service," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4037–4050, 2021.

[18] Z. Li, Z. Yang, P. Szalachowski, and J. Zhou, "Building low-interactivity multifactor authenticated key exchange for industrial internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 844–859, 2021.

[19] J. Yu and R. Hao, "Comments on Sepdp: Secure and Efficient Privacy Preserving Provable Data Possession in Cloud Storage," *IEEE Transactions on Services Computing*, 2019.

[20] Y. Liu, S. Zhang, J. Zhang, L. Tang, and Y. Bai, "Assessment and comparison of six machine learning models in estimating evapotranspiration over croplands using remote sensing and meteorological factors," *Remote Sensing*, vol. 13, no. 19, p. 3838, 2021.

[21] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: survey and open issues," *IEEE Access*, vol. 6, Article ID 18209, 2018.

[22] S. Taghavi and W. Shi, "Edgemask: an edge-based privacy preserving service for video data sharing," in *Proceedings of the 2020 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 382–387, IEEE, San Jose, CA, USA, November 2020.

[23] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, "A scalable and privacy-aware iot service for live video analytics," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, Taipei, Taiwan, June 2017.

[24] J. Gu, J. Wang, Z. Yu, and K. Shen, "Traffic-based side-channel attack in video streaming," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 972–985, 2019.

[25] T. Warren Liao, "Clustering of time series data-a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[26] L. Ng and R. Ng, "On the marriage of lp-norms and edit distance," *Proceedings 2004 VLDB Conference*, vol. 30, pp. 792–803, 2004.

[27] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, vol. 69–84, 2007.

[28] T. Eiter and H. Mannila, "Computing Discrete Fréchet Distance," Technische Universitat Wien, Vienna, Austria, CD-TR 94/64 tech. rep, 1994.

[29] B. Su and J. Su, "One way distance: for shape based similarity search of moving object trajectories," *GeoInformatica*, vol. 12, no. 2, pp. 117–142, 2008.

[30] X. Ding, K. Hao, X. Cai, X. S. Tang, L. Chen, and H. Zhang, "A novel similarity measurement and clustering framework for time series based on convolution neural networks," *IEEE Access*, vol. 8, Article ID 173158, 2020.

[31] M. Wang, X. Tang, F. Chen, and Q. Lu, "Encrypted live streaming channel identification with time-sync comments," *IEEE Access*, vol. 10, Article ID 27630, 2022.

[32] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018.

[33] A. Reed and B. Klimkowski, "Leaky streams: identifying variable bitrate dash videos streamed over encrypted 802.11 n connections," in *Proceedings of the 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, January 2016.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1059997, 9 pages
https://doi.org/10.1155/2022/1059997

Research Article

A CFL-Based Key Management Scheme for Routing-Driven Internet of Things

Jiuru Wang, Chongran Sun, Haifeng Wang, Bin Zhao, and Ping Gong

School of Information Science and Engineering, Linyi University, Linyi, Shandong 276005, China

Correspondence should be addressed to Bin Zhao; jnzhaobin@163.com

Received 7 January 2022; Revised 10 March 2022; Accepted 22 March 2022; Published 26 April 2022

Academic Editor: Zheng Yang

This study is aimed at the authentication problem between the node's public key and the node in the sensor network of the Internet of Things(IoT). As well as the sensor node key distribution needs to verify trusted nodes, resulting in a lot of storage and computational overhead problems. A routing-driven key management scheme for the IoT based on identification certificate authentication system is proposed. The scheme takes the identification key pair of the node as the verification key to verify the random key pair generated by the node to ensure that the whole process does not need the intervention of a trusted third party; random key pairs are generated by nodes independently to ensure that each sensor node has different keys. When a node is broken, it will not cause damage to other nodes. At the same time, the shared key is only established for adjacent sensor nodes that communicate with each other to ensure the security and lightweight storage overhead of the sensor nodes. The experimental analysis shows that the scheme can provide better security, can effectively reduce sensor nodes' storage space and energy consumption, and has higher advantages in safety and performance.

1. Introduction

The mobile IoT is an important part of the new infrastructure. Promoting the innovative development of the mobile IoT will help accelerate the digital transformation of traditional industries and further support the construction of manufacturing power and network power. Applying the technology of the Internet of Things to the construction of smart cities will help improve the urban structure and enhance the city's comprehensive service level [1]. The essential thing for building a smart city is the sensor network (HSN), composed of different types of sensor nodes as the sensing layer of the IoT. Ensuring safe communication between sensor nodes is a priority issue for the IoT [2]. Therefore, key management aiming to provide secure and reliable communication is the most critical and essential content of IoT security research [3]. The research on key management based on wireless sensor networks has achieved many results in traditional networks. However, due to the limited resources of IoT nodes, lack of infrastructure support, and the vulnerability of deployment environments, many existing research results (Such as PKI/CA technology) cannot be directly applied [4].

In recent years, the proposed key management scheme has been mainly based on the public key cryptosystem authentication method. Its core solution lets each node share a key with its neighboring nodes to achieve secure communication [5]. Choi et al. [6] proposed an encryption strategy based on geographic location information that only relies on node location information. The strategy does not need to know the specific deployment information of the node and considers the attack situation inside and outside the node but does not consider the problem of node identity authentication, resulting in poor security. Zhang et al. [7] and others proposed a lightweight asymmetric group key agreement protocol between clusters, which established a secure and efficient group communication channel for sensor nodes between clusters. Elhoseny [8] et al. proposed a key management scheme of elliptic curve cryptography algorithm and homomorphic encryption algorithm based on asymmetric public key cryptosystem, which reduced the cost of communication, memory, and energy. Alappatt and

Prathap [9] mixed Diffie-Hellman key exchange and Elliptic Curve Cryptography (ECC) methods so that each cluster head in the cluster keeps the public key of its corresponding member node and only acts as a router. Mesmoudi et al. [10] proposed a dynamic key management scheme for HSN to ensure the scalability and flexibility of the network.

From the above key management scheme analysis, it can be seen that in the public key cryptosystem, the binding between the public key and the node is mainly based on certificate authentication and identity authentication. However, there are still some defects in these two authentication methods: the node public key (PK) has nothing to do with the node identification in certificate authentication; it needs to be proved by a trusted third party [11]. In identity authentication, the node key is entirely generated by the Key Management Center (KMC), and the node has no complete control over its private key (SK). Furthermore, in the HSN network, to ensure the connectivity of the whole IoT, each node needs to store a sizeable key pool, resulting in colossal storage and computing pressure. Especially when the sensor network is connected to other external networks (including the Internet), it is easy to be attacked by external networks. Once the attacker obtains the information of the key pool from the captured node, the entire network's security may collapse.

This paper aims at the defects of the public key authentication way of sensor nodes in the existing key management schemes and the fragile security and limited resources of sensor nodes in IoT networks. Combined with the identification-based certificate authentication scheme CFL (C, F, and L are the first letters of the last name of three inventors, Chen Huaping, Fan Xiubin, and Lv Shuwang), a route-driven key management scheme based on CFL is analyzed and constructed. The scheme only establishes the shared key for the adjacent sensor nodes that may communicate with each other and uses the CFL authentication system to select the shared key. The identification key pair of the node is used as the verification key pair to verify the random key pair generated by the node. At the same time, the random key pair makes each sensor node have different keys. When a node is broken, it will not cause damage to other nodes. The whole process does not need the intervention of a trusted third party to ensure the security and lightweight storage overhead of the sensor node. An efficient and trusted scheme is proposed to solve the security problems of authentication and communication of node and public key binding in the IoT.

2. Preliminaries

2.1. Identity-Based Certificate Authentication System CFL. CFL is a new identification-based authentication system, which combines certificate authentication and identity authentication. The CFL certification system was first introduced in 2011 and approved by the National Password Administration in 2016 [12, 13]. This scheme combines Public Key Infrastructure (PKI) and Identity-based Cryptography (IBC) authentication schemes to overcome the shortcomings of the existing authentication schemes, making this algorithm a self-authentication authentication algorithm. It achieves centralized key management and guarantees the user's ownership of the random private key. It can meet the security needs of users in public networks to protect their privacy.

The basic key pair of this scheme consists of an identity key pair and random key pair. User ID generates identity key pair as certificate signature and verification key pair and provides certificate signature and verification for user-generated random key pair. A certificate authentication system with a self-certification function is formed, and the whole verification process does not require the intervention of trusted third parties. The specific steps shown in Figure 1 include the following:

(1) User

① Generate real identity ID and generate a random set of public and private key pairs (RAPK, RASK) according to the selected working password algorithm.

② Submit the user identity ID and the self-generated random public key RAPK to the Key Management Center (KMC). The KMC is a credit endorsement that certifies that the ID and signature are generated by a trusted institution.

(2) Key Management Center (KMC)

③ KMC reviews the submitted identification ID to ensure its authenticity and uniqueness

④ Generate an identification key pair (IDPK, IDSK) according to the submitted information

⑤ Use IDSK as the key to sign the certificate with RAPK as the core content and issue the signed certificate to the user

(3) Public side

⑥ Use IDPK as the public key of the verification algorithm to verify the signed certificate. If the verification is correct, the certificate is passed; otherwise, it is not passed.

2.2. Route-Driven Key Management Scheme. In addition to many ordinary sensor nodes (L-Sensors), there are also some special wireless communication devices (H-Sensors) with strong storage and computing power. HSN uses the many-to-one communication mode in the IoT to divide the sensor nodes into several clusters through a clustering algorithm. Each cluster contains a high-energy node and multiple ordinary nodes (shown in Figure 2). Among them, the high-energy node, also known as the cluster head, is responsible for controlling the normal operation of a cluster [14]. The *L* sensing node is responsible for collecting information from the surrounding environment and sending the collected information to the H sensing node through multihop communication [15]. Each sensor node transmits the data to the cluster head node in the cluster. After data fusion, the cluster head node uses multihop communication to send the data to the sink node (sink).

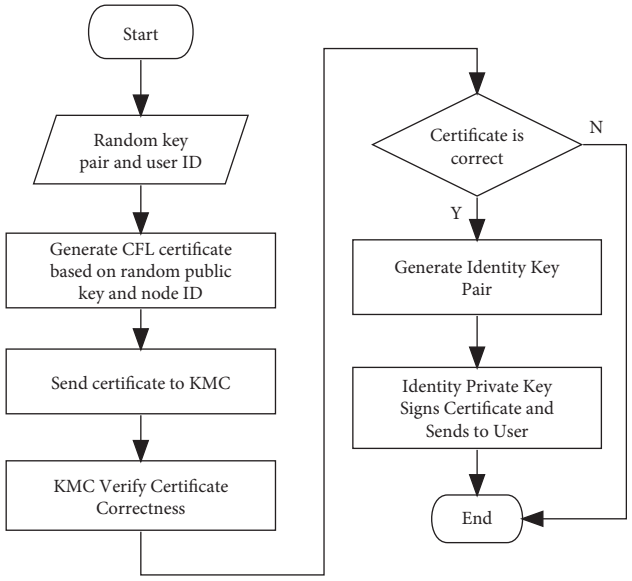


Figure 1: Authentication steps based on CFL.

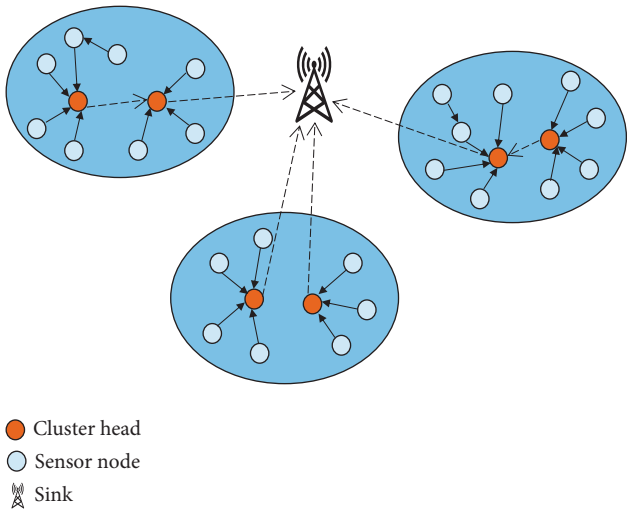


Figure 2: Heterogeneous IoT structure.

In reality, sensor nodes of many-to-one traffic mode only communicate with a small number of adjacent nodes on the transmission path [16], which means nodes do not need to communicate with all neighbors. Therefore, the route-driven key management scheme [17] only sets the shared key for each sensor node and the first adjacent node in the path where its data reaches the receiving node; it is unnecessary to establish the shared key for each neighbor sensor node. When the sensor node sends a packet to the cluster head, the packet will be forwarded by other sensor nodes in the cluster. The intracluster routing scheme determines the node through which the packet needs to pass from the sensor node to its cluster head by making all sensor nodes form a minimum spanning tree (MST) or shortest path tree (SPT) with the cluster head as the root. In order to construct the routing structure, the cluster headfirst needs to obtain the location information of each sensor node. Then the cluster head uses a centralized algorithm to construct the spanning tree according to the relative location of each sensor node. After the routing information is determined, the cluster head uses one or more broadcasts to propagate the routing structure (parent-child relationship) to all sensor nodes.

After the routing structure is determined, the cluster head encrypts the shared key (Symmetric Cipher) between the parent and child nodes through the elliptic curve algorithm (ECC). It sends the key information to the corresponding node. After receiving the message, the node decrypts it to obtain the shared key between itself and its neighbor nodes. It uses the shared key to establish secure communication between adjacent nodes. Each sensor node communicates with only a small part of its neighboring nodes in this scheme, which greatly reduces the communication and computing overhead of key settings. At the same time, the symmetric cryptographic algorithm is used as the shared key between adjacent nodes, which reduces the storage requirements of each sensor node. But correspondingly, because nodes are deployed in dangerous environments, the security of node information cannot be guaranteed.

3. CFL-Based Routing-Driven Key Management Scheme

In this section, a key management scheme for HSN is proposed. The scheme uses the CFL authentication system and many to one communication mode in the HSN network, called route-driven IoT key management scheme based on CFL.

3.1. Scenario Assumptions. In order to focus the research on the algorithm design of route driven key management scheme based on CFL, the paper makes the following assumptions:

(1) The L-sensor and H-sensor nodes are evenly and randomly distributed in the network.

(2) The network is a two-dimensional plane. After the sensor nodes are deployed, an efficient clustering algorithm is used in HSN to form a cluster [18]. Each L-sensor node selects the closest H-sensor node as the cluster head (unless there are obstacles between them). After the cluster is formed, the HSN is divided into multiple clusters, in which the H-sensor node acts as the cluster head to form the backbone of the HSN network.

(3) Each H-sensor node can communicate directly with adjacent H-sensor nodes (if not, the H-sensor node relays through the L-sensor node [19]).

(4) Each L-sensor and H-sensor node has a unique node ID and knows its location.

3.2. Routing Structure of HSN. When the hierarchical network architecture in HSN is formed, the routing in HSN includes two stages:

(1) Intracluster routing: each L-sensor node sends the collected data to its cluster head (H-sensor node)

(2) Intercluster routing: each cluster head integrates the data from multiple L-sensor nodes and sends the data to the receiving node through the network backbone

When the L-sensor node sends a packet to its cluster, the packet will be forwarded by other L-sensor nodes in the cluster. The routing structure in the cluster determines the node through which the packet passes when the L-sensor node transmits the packet to its cluster head. The basic idea of establishing the routing structure in the cluster is to make all l sensing nodes in the cluster form a tree with the cluster head as the root.

In this model, adjacent sensor nodes in the same cluster generate the same data structure (all packets generated by adjacent sensor nodes are k-bit), which is shown in the literature [20]:

(1) The MST consumes the least total energy in the cluster when the intermediate node performs a complete data fusion (i.e., two k-bit packets that become one k-bit packet) during data forwarding.

(2) If there are no data fusion in the cluster, the shortest path tree (SPT) consumes the least total energy.

(3) For partial data fusion, finding the tree with the least total energy consumption is an NP-complete problem. Therefore, MST is used to construct the routing structure in this paper.

In order to build MST, each L-sensor node sends location information to the cluster header H and then constructs a routing structure based on the relative location of each L-sensor node. When the MST construction is complete, the cluster header uses broadcasting to send the tree structure (parent-child relationship between L-sensor nodes) to all L-sensor nodes. It is important to note that broadcasts from cluster headers need to be authenticated [21] to avoid malicious broadcasts by attackers that can disrupt the dissemination of routing information. Authentication of broadcasting identity will be discussed in the next section.

Because L-sensor nodes are small, easily captured, unreliable devices, and may time out and fail [22], MST or SPT algorithms will find multiple parent nodes for each L-sensor node when determining the routing structure. A parent node acts as the primary parent node, while other parent nodes act as the standby parent node. When the primary parent node fails, the L-sensor node uses the standby parent node for routing without crashing the entire communication network due to a failure of one node. Once the routing structure is determined, each L-sensor node only needs to establish a shared key with its parent and child nodes.

3.3. Key Distribution. In this section, we describe a shared key distribution scheme between adjacent sensor nodes based on CFL. The basic cryptographic algorithms for random key pairs can choose exponential product public-key cryptography, elliptic curve cryptography ECC, or RSA algorithms. In this section, only SM2- and SM3-based key management schemes are introduced, and the process is described as follows:

(1) The L-sensor node generates a set of random keys (RASK, RAPK) based on a previously selected working password algorithm.

(2) After the cluster head selection is completed, the L-sensor node u (Lu) sends the random public key RAPK_u and node information identification ID_u to the cluster head, which is designated as $\mathrm{ID}_1 = \mathrm{RAPK}_u || \mathrm{ID}_u$. At the same time, since the location of the cluster head is known to all common sensor nodes in the cluster during cluster formation, the greedy geographic routing protocol [23] ensures that the information is forwarded to the cluster head.

(3) When the cluster head receives the information from the L-node, it calculates $h = H(\mathrm{ID}_i) = \{h_0, h_1, \ldots, h_{t-1}\}$, h_i: $\{i = 0,1, \ldots, t-1\}$ to get the control information h from the input of the control function. Where H uses the password hash function SM3, the output is $N = 256$ bits, and $N = st$ is set, s is the key length and meets $s \mid N$. The cluster head calculates the multilinear control function according to h as follows:

$$f_h(SKB) = f_h(sk0, sk1, \ldots, sk2s - 1 = \mathrm{IDSK}). \quad (1)$$

Generate node identity private key($IDSK$). SKB is the private key base, and the generation method is as follows: make m the period of base point P in SM2, cluster head randomly selects $sk_i \in Zm = Z/mZ$, $i = 0,1,\ldots,t2s-1$, and the two are not equal to each other to get the private key base:

$$\mathrm{SKB} = (sk0, sk1, \ldots sk_{t2s-1}). \quad (2)$$

(4) The cluster head constructs an MST based on the location information of the L node and centralized MST algorithm to get the L_v of the parent node of L_u. Then use the signature algorithm SIGN, with IDSK as the key, to sign the contents of the certificate, which is recorded as $\mathrm{sign}_u = \mathrm{SIGN}_{\mathrm{IDSK}}(\mathrm{ID}_u || \mathrm{RAPK}_u || v$ (parent node of u)$||$ RAP K_v, send the signed certificate sign_u to L_u.

(5) After obtaining the certificate sign_u, L_u inputs ID_1 into the cryptographic hash function SM3 to obtain the control information h input by the multilinear control function. According to h and the public key generator, it is transformed by the following multilinear function:

$$f_h(\mathrm{ID}, \mathrm{PBK}) = f_h(pk_0, pk_1, \ldots, pk_{t2s-1}) = \mathrm{IDPK}. \quad (3)$$

PKB is the public key generator, PKB=(pk_0, $pk_1,\ldots,pk_{t2s-1}$), and $pk_i = sk_i \cdot P \bmod E$, gets the public key base. Generate the identity public key IDPK_u, and use IDPK_u as the public key of the verification algorithm to verify the signed certificate.

(6) After the verification is correct, the certificate is passed. L_u uses the public key *RAP* K_v of the parent node L_v to communicate with L_v.

The pseudo-code of key distribution algorithm is shown in Algorithms 1–3.

3.4. Key Revocation

3.4.1. Cluster Member Revocation. When the L-sensor node in the cluster is destroyed, it is necessary to revoke all keys about the L-sensor node and update the routing structure about the node. When the cluster head node detects that the node is damaged, the cluster head node determines according to the position of the damaged node in the routing structure:

(1) When the node to be revoked is a leaf node, only one revocation information needs to be sent to its parent node

(2) When the node to be revoked is the parent node, and its child node has a standby parent node, send the revocation information to the parent node and child node of the damaged node, and use the standby parent node for communication in the next communication

(3) When the node to be revoked is the parent node, and its child node has no standby parent node, the cluster head needs to re-establish the route for the child node of the damaged node and pass the revocation information and the new routing structure to the child node

The revocation message contains the key list to be revoked (symmetric key for communication between nodes). The key list is signed with the identification private key IDSK(expressed as sign), and the sign is appended to the key list. Each L-sensor node has a separate identification public–private key pair. Therefore, when the L-sensor node receives the revocation message, it verifies the digital signature through the identification public key IDPK to check the integrity and authenticity of the message. It can effectively prevent the opponent from sending false revocation messages.

3.4.2. Cluster Member Revocation. Like cluster members, cluster heads are also hostile and need to be adjusted when they are captured or damaged. When the base station detects that the cluster head is captured or damaged, it queries all the node information in the cluster based on the identity of the cluster. It broadcasts the updated information to all the nodes in the cluster. After receiving the updated information, the cluster members delete the existing key pairs and query the nearest cluster head information except for the original cluster head, apply to join the cluster, redistribute the key, and form the IoT network with the new cluster head.

4. Experiment and Analysis

4.1. Safety Analysis

Theorem 1. *CFL is a computationally difficult and provable security scheme.*

Prove. The working private key of the CFL system user and the signature private key generation meta-set of the CFL Certificate Generation Center are independent of each other. Only the random public key of the L node and the signature verification public key generation meta-set of the H node are published in the key work phase. Therefore, in theory, the attacker's attack on the original set of the signature private key generated by the certificate generation center and the working private key of the CFL user is transformed into a corresponding mathematical problem.

Theorem 2. *If the adversary breaks through a cluster member and obtains the random private key and signature public key of the node, it does not affect the private key of other cluster members and the private key generation meta set of the cluster head.*

Prove. Probability Turing Machine TM $(M, \sum)$ given polynomial time, where M is signature information, E is a signature random variable, and $M, \sum$ is independent of each other, $(M, \sum)$ induces random variable $(H, M, \sum)$, where H is a Hash function; then,

$$\forall h, \sigma, \Pr((H(M), \Sigma) = (h, \sigma)) = \frac{1}{2^{2n}}. \tag{4}$$

In the CFL certification system, $\forall h$ makes

$$\Pr((H(M), \Sigma) = (h, \sigma)) = \begin{cases} \frac{1}{2^n}, & Sign_{CFL}(h) = \sigma \\ 0, & else \end{cases}, \left|\Pr((H(M), \Sigma) = (h, \sigma)) - \Pr\left((H(M), \Sigma') = (h, \sigma)\right)\right| \le \frac{1}{2^n}. \tag{5}$$

So in polynomial time,

$$\sum_{|(h,\sigma)| < n^c} \left|\Pr((H(M), \Sigma) = (h, \sigma)) - \Pr\left((H(M), \Sigma') = (h, \sigma)\right| \le \frac{n^c}{2^n}. \tag{6}$$

(1) IF L-Node U applies for registration THEN
(2) Upload U identity information
(3) IF Authentication pass THEN
(4) $RAPK_u$, $PASK_u \leftarrow$ Generate
(5) $\mathrm{ID}_u \leftarrow$ Node ID
(6) $\mathrm{ID}_1 \leftarrow \mathrm{RAPK}_u \| \mathrm{ID}_u$
(7) H-Node $H \leftarrow \mathrm{ID}_1$
(8) END IF
(9) END IF

Algorithm 1: Node operation algorithm.

(1) MST $\leftarrow$Generate
(2) //Generate MST according to node location information
(3) $L_v \leftarrow$ Generate
(4) //Generate parent node of Lu according to MST
(5) IDSK $\leftarrow f_h$ (SKB)
(6) //SKB = $(sk_0, sk_1, \ldots, sk_{t2s-1})$; $h = H(\mathrm{ID}_i) = \{h_0, h_1, \ldots, h_{t-1}\}$
(7) $\mathrm{sign}_u \leftarrow \mathrm{SIGN}_{\mathrm{IDSK}}(\mathrm{ID}_u \| \mathrm{RAPK}_u \| (v(\text{parent node of } u) \| \mathrm{RAP}\ K_v)$
(8) $\mathrm{L}_u \leftarrow \mathrm{sign}_u$

Algorithm 2: Cluster head operation algorithm.

(1) IDPK $\leftarrow f_h$ (ID, PBK) //PKB = $(pk_0, pk_1, \ldots, pk_{t2s-1})$, $pk_i = sk_i\ P \bmod E$
(2) IF $\mathrm{VERIFY}_{\mathrm{IDPK}}$ (sign_u) = TRUE
(3) L_u *and* L_v use RAPK_v to communicate
(4) ELSE
(5) Abandon sign_u
(6) End IF

Algorithm 3: Node operation algorithm.

Let the set of nodes be U; if $\forall u_1, u_2 \in U$, and $u_1 \neq u_2$,

$$P\left(\mathrm{RASK}_{u_1} = \mathrm{RASK}_{u_2}\right) < \varepsilon. \tag{7}$$

The node private key satisfies one-node-one-secret if $\varepsilon \longrightarrow 0$.

Thus, when SM2 and SM3 satisfy random predictors, the CFL certification system based on SM2 and SM3 satisfies statistical zero-knowledge. The random keys of the cluster members and the signature verification cipher algorithm of the cluster head satisfy "one-node-one-secret" in use [24]. When an attacker gets the key pair information of a node, the adversary can get the key length s of the node to guess the keys of other nodes with a guess space of 2^s. Only when an attacker obtains the signature private key of $t2^s$ nodes can the private key generation meta set of the cluster head node be obtained. However, the cost of obtaining a token private key from a token public key is enormous, and the difficulty of obtaining $t2^s$ token private keys is not calculable under current computing conditions.

Theorem 3. *In this scheme, the adversary cannot break the entire key system by intercepting the cluster head.*

Prove. If the adversary captures a cluster head, the base station can detect and broadcast the revocation message of the cluster head. All cluster members of the original cluster head delete the random key and flag key pairs used for communication within the existing cluster, find the closest H node except the original cluster head and apply to become a member within the cluster. The routing structure and node key of the original cluster head are invalid. When a cluster member joins a new cluster, his leaf node key is modified, so the adversary cannot obtain the communication key of the newly joined cluster.

The three theorems in this section prove that the key leakage of a single node in the system does not affect the security of other nodes and the whole system. It shows that the scheme has high antinode acquisition ability and security and ensures the security of cluster heads and cluster members in the adversary environment.

4.2. Performance Analysis. Because most sensor nodes have limited resources, in addition to security, the storage requirements, connectivity, computation, and energy consumption of the key management algorithm are important

Table 1: Algorithm complexity analysis.

Protocol	Storage space	Connectivity	Tate pairing	Length of message sent	Length of message received
[7]	$(M+4)N+5M$	1	5	$4\|G_1\|$	$(N+4)\|G_1\|$
[17]	$(M+2)N+3M$	1	2	$2\|G_1\|$	$3\|G_1\|$
Ours	$M*N+3N$	1	1	$3\|G_1\|$	$4\|G_1\|$

Storage space indicates the capacity required by the sensor to store keys. Connectivity represents the connection efficiency of sensors in different protocols. Tate pairing refers to bilinear pairing during key generation.

indicators to measure the performance of the protocol [5]. This paper compares the key management scheme used in wireless sensor networks with this scheme. Table 1 lists the comparison results between this scheme and the comparable key management scheme in terms of storage requirements, computational complexity, and connectivity.

Assume that the number of H-sensors and L-sensors in observation area is M and N, respectively. Typically, we have $M << N$. In the CFL-based key management protocol for the IoT, the H-sensors store the node information certificates $ID_L = RAPK_L || ID_L$ of all nodes in the cluster (including the random public key $RAPK_L$ and the node information identification ID_L), the L-sensors store the self-generated random private key $RASK_L$ and the identification public key $IDPK_L$ and the random key RAP K_v of the parent node required to communicate with neighbor nodes, so the storage requirements for this solution are

$$M * N + 3N. \tag{8}$$

In the ECC-based key management scheme [17], H-sensors need to preload the public keys of all L-sensors, a pair of their public and private keys, and a deployment key K_H; L-sensors need to preload its private key and H's public key, so the total number of keys preloaded by ECC-based key management scheme is

$$M * (N + 3) + 2N = (M + 2)N + 3M. \tag{9}$$

In scheme [7], H-sensors need to store intercluster Federation key, node information (session key identity, partner identity, group session key pair), and intracluster node information. In contrast, L-sensors only need to store their node information. The required storage space is

$$5M + MN + 4N = (M + 4)N + 5M. \tag{10}$$

Figure 3 shows that the schema [7] requires the highest storage space. Scheme [17] requires very close storage space with the same number of nodes, but its key generation still needs the key management center, and the node does not have full control over the private key.

From the energy consumption of the protocol, this paper quantifies the total energy consumed by the key management scheme to the sum of the calculation and communication consumption of the group members in the negotiation process, which is general. According to the data provided in [25], the energy consumption of a 133 MHz "Strong ARM" microprocessor for computing and communication is shown in Table 2:

Since the security of the 160-bit ECC encryption algorithm is comparable to that of 1024-bit RSA and DSA

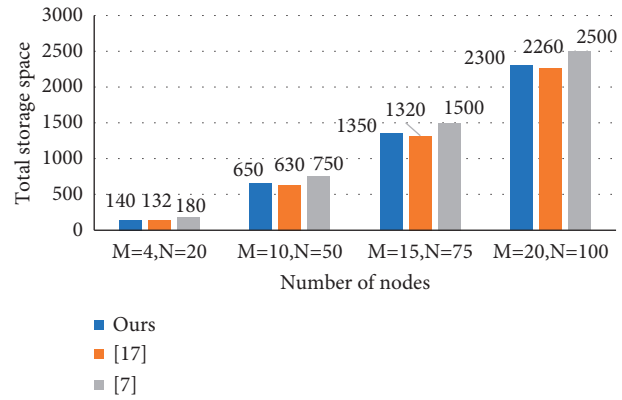


Figure 3: The relationship between node number and total storage demand.

Table 2: Energy cost for computation and communication.

Type of communication	Energy cost/mJ
Computation cost of tate pairing	47.0
Communication cost for sending 1bit	0.66×10^{-3}
Communication cost for receiving 1bit	0.31×10^{-3}

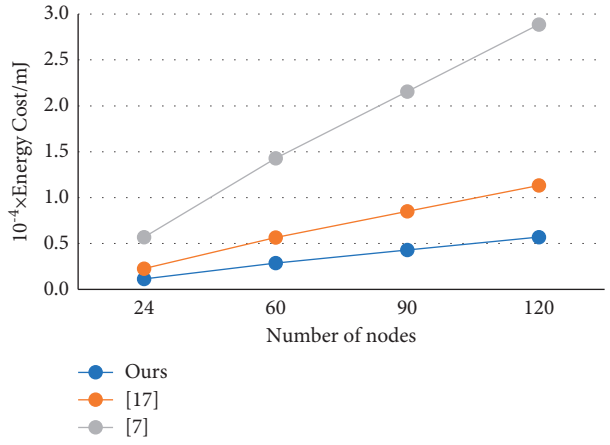


Figure 4: Total energy consumption.

encryption algorithm [26], it is assumed that the single information quantity of key management algorithm based on elliptic curve encryption scheme is $|G_1| = 160$ bit. Then the total energy consumption analysis is shown in Figure 4. The total energy consumption of the scheme [7] increases rapidly with the increase of the number of nodes; The scheme in this paper has a significant advantage over the scheme [17] in terms of total energy consumption.

Moreover, this scheme has fewer storage requirements, greater flexibility, and higher security.

5. Conclusions and Future Work

This paper combines identity-based certificate authentication system CFL with heterogeneous sensor networks. It proposes a new routing-driven key management scheme based on CFL to solve the problems faced by current heterogeneous IoT key management. It effectively solves the authentication and communication between the nodes and public keys of the IoT. At the same time, third-party services are not required to participate in the key establishment process. Key information will not be leaked during the key transfer process to ensure the security of the key transfer. The final result analysis shows that this scheme has obvious advantages in security, storage requirements, connectivity, and energy consumption and is more suitable for low-configuration wireless IoT networks. The next step is to further research and realize its key management application in decentralized [27] (such as blockchain) IoT application scenarios.

Data Availability

The data used to support the results of this study, including algorithms and proofs, are included in this paper and can also be obtained from the corresponding authors upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially funded by the Key Research and Development Program of Shandong Province (2019GNC106027 and 2019JZZY010134) and the Natural Science Foundation of Shandong Province (ZR2020MF058 and ZR2020MF029).

References

[1] General Office of Ministry of Industry and Information Technology, "Notice of the general office of the ministry of industry and information technology on deepening the all-round development of mobile Internet of things [EB/OL]," 2020, http://www.gov.cn/zhengce/zhengceku/2020-05/08/content_5509672.htm, in Chinese.

[2] W. Li, D. Wang, and P. Wang, "Insider attacks on multi factor authentication protocol in wireless sensor networks," *Journal of Software*, vol. 30, no. 08, pp. 2375–2391, 2019, in Chinese.

[3] A. Irshad, S. A. Chaudhry, Q. Xie et al., "An enhanced and provably secure chaotic map-based authenticated key agreement in multi-server architecture," *Arabian Journal for Science and Engineering*, vol. 43, no. 2, pp. 811–828, 2018.

[4] A. Karrothu and J. Norman, "Group and hierarchical key management for secure communications in Internet of Things," *International Journal of Communication Systems*, vol. 33, no. 13, p. e3859, 2020.

[5] A. K. Gautam and R. Kumar, "A comprehensive study on key management, authentication and trust management techniques in wireless sensor networks," *SN Applied Sciences*, vol. 3, no. 1, pp. 1–27, 2021.

[6] J. Choi, J. Bang, L. H. Kim, M. Ahn, and T. Kwon, "Location-based key management strong against insider threats in wireless sensor networks," *IEEE Systems Journal*, vol. 11, no. 99, pp. 494–502, 2017.

[7] Q. Zhang, Y. Gan, R. Wang, Z. Jiamin, and T. Yu'an, "Asymmetric group key agreement protocol between clusters," *Journal of Computer Research and Development*, vol. 55, no. 12, pp. 2651–2663, 2018, in Chinese.

[8] M. Elhoseny, H. Elminir, A. Riad, and X. Yuan, "A secure data routing schema for WSN using elliptic curve cryptography and homomorphic encryption," *Journal of King Saud University - Computer and Information Sciences*, vol. 28, no. 3, pp. 262–275, 2016.

[9] V. Alappatt and P. M. J. Prathap, "Hybrid cryptographic algorithm based key management scheme in MANET," *Materials Today Proceedings*, 2020, In Press.

[10] S. Mesmoudi, B. Benadda, and A. Mesmoudi, Skwn, Smart and dynamic key management scheme for wireless sensor networks," *Communication Systems*, vol. 32, pp. 1–23, 2019.

[11] Q. Zhang, X. Hu, W. Liu, and W. Jianghong, "An improved three-party password verification meta-authentication key exchange protocol," *Journal of Software*, vol. 31, no. 10, pp. 3238–3250, 2020.

[12] H. Chen, X. fan, and S. Lu, *Identity-based Certification Scheme CFL*, CN102957536A, Beijing, 2013, in Chinese.

[13] State Password Administration, "Notice on approval of CFL authentication system based on SM2 and SM3 [EB/OL] ," 2016, https://sca.gov.cn/sca/xxgk/2016-03/21/content_1002812.html, in Chinese.

[14] H. Chan, P. Adrian, and D. Song, "Random key predistribution schemes for sensor networks," *2003 Symposium on Security and Privacy*, IEEE, 2003.

[15] A. Chanda, P. Sadhukhan, and N. Mukherjee, "Key management for hierarchical wireless sensor networks: a robust scheme," *EAI Endorsed Transactions on Internet of Things*, vol. 6, no. 23, 2020.

[16] P. Li and L. Zhu, "A multi-conditional proxy broadcast Re-encryption scheme for sensor networks," *Computers, Materials & Continua*, vol. 65, no. 3, pp. 2079–2090, 2020.

[17] X. Du, Y. Xiao, S. Ci, M. Guizani, and H. Chen, "A routing-driven key management scheme for heterogeneous sensor networks," in *Proceedings of the 2007 IEEE International Conference on Communications*, pp. 3407–3412, IEEE, Glasgow, UK, June 2007.

[18] Z. Kuang, J. Wu, and J. Li, "Ring KNN algorithm based on clustering," *Computer engineering and Science*, vol. 41, no. 05, pp. 804–812, 2019, in Chinese.

[19] X. Wang, H. Wu, D. Liu, D. Ye, and Z. Yang, "A global key management method for hierarchical wireless sensor networks," *International Core Journal of Engineering*, vol. 6, no. 7, pp. 327–333, 2020.

[20] D. Nageswari, R. Maheswar, and G. R. Kanagachidambaresan, "Performance analysis of cluster based homogeneous sensor network using energy efficient N-policy (EENP) model," *Cluster Computing*, vol. 22, no. 5, pp. 12243–12250, 2019.

[21] C. Wang, D. Wang, G. Xu, and D. He, "Efficient privacy-preserving user authentication scheme with forward secrecy

for industry 4.0," *Science China Information Sciences*, vol. 65, no. 1, 2022.

[22] C. Esposito, M. Ficco, A. Castiglione, F. Palmieri, and A. D. Santis, "Distributed group key management for event notification confidentiality among sensors," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 566–580, 2018.

[23] Q. Xian and Y. Long, "An enhanced greedy perimeter stateless routing algorithm for wireless sensor network," in *Proceedings of the 2016 IEEE international conference of online analysis and computing science (ICOACS)*, pp. 181–184, IEEE, Chongqing, China, May 2016.

[24] C. Du, J. Liu, and X. fan, "CFL satisfies statistical zero-knowledge," *Research on Information Security*, vol. 2, no. 007, pp. 621–627, 2016, in Chinese.

[25] Q. Zhang, R. Wang, and Y. Tan, "Identity-based verifiable asymmetric group key agreement protocol," *Journal of Computer Research and Development*, vol. 51, no. 8, pp. 1727–1738, 2014.

[26] L. I. Zengpeng, V. Sharma, M. A. Chunguang, G. E. Chunpeng, and S. U. S. I. L. O. Willy, "Ciphertext-policy attribute-based proxy re-encryption via constrained PRFs," *Science China(Information Sciences)*, vol. 64, no. 06, pp. 242-243, 2021.

[27] C. Qu, M. Tao, J. Zhang, X. Hong, and R. Yuan, "Blockchain based credibility verification method for IoT entities," *Security and Communication Networks*, vol. 2018, pp. 1–11, 2018.