Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4987002, 10 pages
https://doi.org/10.1155/2022/4987002

Research Article

Key Update and Device Location of Internet of Things Based on Smart Contract

HongQi Bi

YuZhang Normal University, Nanchang, Jiangxi 330103, China

Correspondence should be addressed to HongQi Bi; bihongqi@yuznu.edu.cn

Received 22 December 2021; Revised 13 January 2022; Accepted 28 January 2022; Published 17 March 2022

Academic Editor: Zhiguo Qu

In order to reduce the key update time delay of the Internet of Things and reduce the device location error rate and location time, a new key update and device location of Internet of Things based on smart contract was designed. Adopt improved MVIF security mechanism of both parties to build smart contract security mechanism, and the key update of the Internet of Things is realized through key predistribution, user registration and login, and key update. On this basis, multidimensional scaling technology is used to abstract Internet of Things devices into relative coordinates in multidimensional space, and the absolute coordinates are obtained by eliminating the distance estimation error to achieve Internet of Things device positioning. Experimental results show that the proposed method has a lower key update time delay, a lower amount of data used in the key update process, a lower positioning error rate of Internet of Things devices, and a shorter positioning time, which fully verifies the effectiveness of the proposed method.

1. Introduction

With the development and popularity of the Internet of Things technology, intelligent devices of the Internet of Things have gradually penetrated into all aspects of people's lives. From the development trend of intelligent devices of the Internet of Things in recent years, the number of intelligent mobile devices represented by smart phones has been in a state of rapid growth [1]. Ensuring the security and credibility of information is an important prerequisite to ensure good user experience. In order to ensure secure communication between intelligent devices, a variety of security algorithms have been proposed [2], and location-based services have also been widely used. In order to achieve secure communication of intelligent devices, efficient secure channels are usually established between devices through symmetric encryption algorithm, and key negotiation and message verification are realized through public key encryption and public key signature algorithm [3, 4].

In order to facilitate users to obtain public key information during the encryption or signature verification process, a new identity-based encryption algorithm is designed. The algorithm uses the user's identity as the public key, and the private key generation center provides the user with the private key. This encryption algorithm makes it unnecessary for the user to apply for a digital certificate, thus avoiding the tedious work of certificate issuance, revocation, verification, and storage [5]. At the same time, identity as a public key greatly improves the efficiency and credibility of public key query. However, in order to prevent attackers from forging or replaying by brute force cracking the key, users often need to update the key, which puts a high demand on the private key generation center, which needs to calculate the latest key for each user and establish a channel with each user for key distribution. Similarly, BeiDou Navigation Satellite System and GLOBAL Positioning System require each user to receive stable positioning signals from satellites when positioning smart devices [6]. However, due to the complexity and uncertainty of the environment, for example, in urban indoor or outdoor, communication signals and positioning signals between smart devices and public infrastructure (private key generation center, positioning satellite, etc.) may be in a weak signal or no signal state,

resulting in very difficult key update and device positioning [7]. Many self-organizing public key protocol management systems have been proposed so far, and devices can distribute and revoke their public keys without the help of any fixed server. However, these systems require that devices be always online when other devices publish their session keys or update their public keys [8]. In addition, during multihop communication, there is no trusted third party that can monitor the behavior of all devices and arbitrate disputes in the presence of malicious devices. In addition, if users need to obtain location information of other smart devices, it is difficult to verify the accuracy of the location information published by existing technologies. Therefore, the problems of self-organizing key update and trusted location for intelligent devices in complex environment have not been solved yet.

To solve the key update problem, reference [9] proposed a security authentication and key update method for multipoint cooperative joint transmission of Internet of vehicles. In this method, the vehicle generates the base station switching request and uses random number, shared key, and target base station public key to encrypt and broadcast the switching request. Based on the characteristics of cryptography, the target base station can not only calculate the shared key from the ciphertext request based on the private key, but also calculate the subsequent session key. The vehicle can calculate the session key based on the location information of the target base station and the random number generated when the request is generated, so as to realize the key sharing and key updating between the vehicle and the base station under the premise of only one key transmission. The key generation and update algorithm is verified and analyzed from the perspective of cryptography. Reference [10] proposed a decentralized identity authentication and key update method based on block chain. The user only needs to generate the identity identifier and key for the first time by IGC (Identity Generator Center), and then the user can automatically update the key. During this process, the identity identifier remains unchanged as the public key. Update only the private key and parameters to facilitate authentication. This paper redefines the data structure of transactions, and the update process is recorded in the blockchain in the form of transactions, which ensures its authenticity and credibility. Since the blockchain data cannot be tampered with, the blockchain data is authentic. Reference [11] proposed a global positioning method for high-value medical devices based on Internet of Things smart tag. Combine new Internet of Things technology with RFID identification, identify equipment tags and extract feature codes of tag positioning data, perform hybrid networking control combined with parameter feature codes, build Internet of Things node deployment models, and use this model to cluster equipment operating data and fusion processing to obtain the device positioning result. Reference [12] proposed a secure positioning method for Internet of Things devices based on blockchain technology, using blockchain distributed ledgers to share between Internet of Things devices. After Internet of Things devices are located, a list of new locations and adjacent nodes is added to the blockchain, and the shared location data is used by other devices to locate them.

In order to reduce the key update time delay of the Internet of Things and reduce the device location error rate and location time, a new key update and device location of Internet of Things based on smart contract was designed. The method realizes key update of the Internet of Things with the support of smart contract security mechanism. Multidimensional scaling technology is used to abstract IoT devices into relative coordinates in multidimensional space, and the absolute coordinates are obtained by eliminating the distance estimation error to achieve IoT device positioning. As the key update and device positioning methods of the Internet of Things are designed separately in this paper, the key update time delay, the amount of data used in the key update process, and the positioning error rate and positioning time of the Internet of Things devices are lower in this paper.

2. Internet of Things Key Update Based on Smart Contract

So far, the development of blockchain technology has gone through three development stages: technology origin, blockchain 1.0, and blockchain 2.0. Among them, the technological origin of blockchain includes asymmetric encryption technology, distributed database technology, point-to-point network technology, and digital currency [13]. In 2009, the digital currency Bitcoin network was officially put into operation. As the underlying technology supporting its operation, it evolved to blockchain 1.0 technology, realizing the combination of technologies such as chain data structure with block as unit, whole-network shared ledger, asymmetric encryption algorithm, and source code and open source [14, 15]. In 2014, blockchain technology began to be applied in other areas than digital currency, such as distributed identity authentication, distributed domain name system, distributed self-organization, and other distributed applications. Blockchain technology evolves to blockchain 2.0, with smart contracts, distributed applications, virtual machines, and other technologies as typical features [16].

Smart contract is a code deployed in a distributed ledger that can realize various business logic functions and is a computer program that can automatically execute the terms of the contract [17–19]. Smart contracts are self-consistent, self-contained, and distributed, defined by code and executed independently. Formalization of the content of the contract is realized by digitally encoding the content of the contract and writing it into the blockchain [20]. In order to meet the security and privacy protection of smart contracts, virtual iterative function (VIF) was used to encrypt the receipts generated by smart contracts in previous studies [21]. Because it adapts to the current contract environment, there are not only bilateral contracts, but also many multiparty contracts. Therefore, this paper takes multidimensional virtual permutation function (MVTF) theory as the research basis, constructs and improves bilateral MVIF security mechanism, and takes it as the smart contract

security mechanism, so as to improve the security of smart contracts. The smart contract construction mechanism is shown in Figure 1.

The smart contract construction mechanism is described as follows:

(1) Multiparty contract users exchange key array mapping security subsystem through token ring to form MVIF space. Token ring can ensure that multiparty users form multidimensional virtual permutation functions in a certain order without omission or repetition. Finally, each user has the same joint key-controlled array, and the security subsystem mapping $n * n$ also has the same MVIF [22, 23].

(2) Multiple users use their respective keys to replace multiple security subsystems in the existing MVIF. The key of each user represents multiple coordinates of one dimension in the MVIF, and the same unique and sequential multiple security subsystems are displaced in MVTF through key exchange [24–26].

(3) The user iterates the security subsystems selected in the previous step in sequence through the compiler to construct the VIF, and the encryption contract obtains the corresponding receipt for saving, which is used for the authentication of the security call and execution of the future contract.

(4) When multiple users contract again, they only need to repeat the above steps to quickly replace a large number of secure encryption methods and apply them to subsequent smart contract authentication receipts [27].

The key update process of Internet of Things based on smart contract is as follows:

(1) Key predistribution

There are roughly two types of devices in the Internet of Things. The first is low-energy device L, which has a large number and a weak storage space for energy and computing power [28]. The second is high-energy equipment, which has great storage space, super computing power and sufficient energy. Since the Internet of Things is focused on a wide range of uses, predeployment is required. The predeployment phase is the startup phase. The system is started by the network administrator and runs offline.

In the predeployment phase, each fixed communication device $\{S_j | 1 \le j \le m\}$ is preassigned its identity information SID_j and key $X_{\mathrm{GWN}-S_j}$, which are allocated by the gateway device (GWN device) and saved by the device itself, where m represents the number of predeployed devices in the Internet of Things [29]. Due to the weak storage computing capacity of general low-energy devices, the small storage space of devices was taken into account at the beginning of the formulation of the scheme [30]. Therefore, in the scheme, the key preservation and predistribution requirements are that the storage space is sufficient. In addition, GWN is pregenerated high security key X_{GWN}, with only the gateway device GWN stored and the gateway device also stores $X_{\mathrm{GWN}-S_j}$ of $\{S_j | 1 \le j \le m\}$ for each device.

In a network-wide preconfigured device key approach, any hazardous device can compromise the security of the entire network fully. In explicit device preconfiguration, although a few dangerous devices are connected to each other, the entire network is not affected. Preconfigured device security methods provide group device protection against threats from other hazardous devices deployed on the same device [31, 32].

(2) User registration and login

The user registration process is as follows:

Step 1: Users carry their own identity information ID_i and password PW_i and select random r_i to carry out password encryption and identity information encryption calculation [33]. The results are as follows:

$$\begin{aligned} \mathrm{MP}_i &= h(r_i \| \mathrm{PW}_i), \\ \mathrm{MI}_i &= h(r_i \| \mathrm{ID}_i). \end{aligned} \tag{1}$$

Step 2: The user sends MP_i and MI_i to the gateway GWN through a private channel. After receiving MP_i and MI_i, GWN encrypts them according to the security key known only to itself and carries out encryption calculation [34]:

$$\begin{aligned} f_i &= h(\mathrm{MI}_i \| X_{\mathrm{GWN}}), \\ x_i &= h\left(\mathrm{MP}_i \| X_{\mathrm{GWN}-U_i}\right). \end{aligned} \tag{2}$$

$X_{\mathrm{GWN}-U_i}$ is the random selection of a security key, f_i is the result of encrypting the mask of user identity information, x_i is the result of encrypting the password of U_i, and then XOR operation [35]

$$e_i = f_i \oplus x_i. \tag{3}$$

Then the network joint GWN stores user MP_i's identity information mask and security key $X_{\mathrm{GWN}-U_i}$.

Step 3: The gateway device takes $\{\mathrm{MI}_i, e_i, f_i, X_{\mathrm{GWN}-U_i}\}$ as SC, the user's identity information card in the Internet of Things, and sends it to the user through the private network. After receiving the packet, the user adds the selected random number r_i to the packet.

After the registration phase, user U_i initiates authentication to connect to any destination device in the Internet of Things. In order to implement user authentication, user login must be implemented first. Based on the above description, smart contract is applied in this paper to realize user registration and authentication, and any other security system for data protection can be replaced by it [36].

Step 1: User U_i enters password WP_i^* according to the smart contract. The smart label is calculated by WP_i^* and the smart label is saved by user r_i.

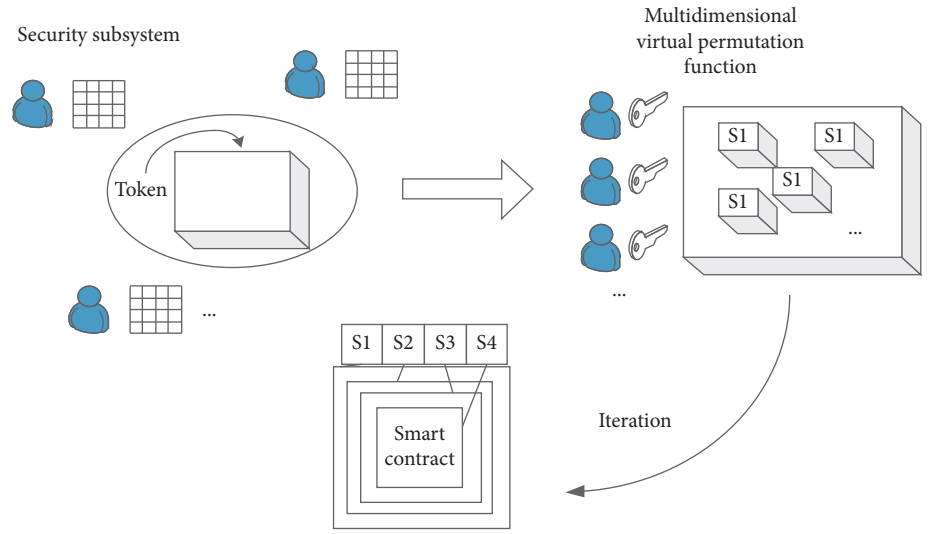

Figure 1: Smart contract construction mechanism.

$$\begin{aligned} MP_i^* &= h(r_i \| PW_i^*), \\ x_i^* &= h\left(r_i \| X_{GWN-U_i}\right). \end{aligned} \tag{4}$$

MP_i is the calculated user mask, and x_i^* is the password result after encryption.

Step 2: Check whether x_i is equal to x_i^*. If the value is different, the user does not enter a correct password during the login. If so, the smart contract performs the following calculations:

$$N_i = h\left(x_i \| X_{GWN-U_i} \| T_1\right). \tag{5}$$

Step 3: Select a random number K_i, calculate $Z_i = K_i \oplus f_i$ according to the smart contract, and send $\{MI_i, e_i, Z_i, N_i, T_1\}$ to destination device S_j over the public channel. The random number K_i is used to construct the shared key for communication with the destination device, and the random number Z_i is used to encrypt the return key.

(3) Key update

For key change stage, the user can freely choose whether or not to change the session key, in the design of the scheme, using the intelligent contract, users must be logged in into the key of change, only after intelligent contract after modifying some parameters, through certification, with equipment and GWN can change in the key be achieve, so at this stage, it does not need to interact with the device or GWN, and this phase is offline.

Step 1: Access the smart terminal, view the smart contract, enter the password PW_i^{OLD}, and calculate MP_i^* based on the smart contract and the saved password r_i.

$$MP_i^* = h\left(r_i \| PW_i^{OLD}\right). \tag{6}$$

In this case, the encryption password is

$$x_i^* = h\left(MP_i^* \| X_{GWN-U_i}\right). \tag{7}$$

Original password:

$$x_i = f_i \oplus e_i. \tag{8}$$

Step 2: Check whether x_i equals x_i^*. If the value is different, the user does not enter a correct password during the login. If so, the smart contract performs the following calculations:

$$MP_i^{**} = h\left(r_i \| PW_i^{NEW}\right). \tag{9}$$

The new password is

$$\begin{aligned} x_i^{NEW} &= h\left(MP_i^{**} \| X_{GWN-U_i}\right), \\ e_i^{NEW} &= f_i \oplus x_i^{NEW}. \end{aligned} \tag{10}$$

Step 3: Replace e_i in the original smart contract with e_i^{NEW}, so that $\{MI_i, e_i, Z_i, N_i, T_1\}$ becomes $\{MI_i, e_i^{NEW}, Z_i, N_i, T_1\}$.

3. Locating Internet of Things Devices

Multidimensional scaling (MDS), as a data analysis method, was originally applied in the field of psychometry to achieve data statistics and analysis in psychological aspects. Multidimensional scaling technology abstracts devices from reality into relative coordinates in multidimensional space. The degree of similarity between devices is related to the degree of Euclidean distance of relative coordinates in multidimensional space.

It is assumed that the dissimilarity of any two Internet of Things devices i and j can be represented by p_{ij}, so the dissimilarity of pairs of devices between the n devices can be represented by a matrix of $[p_{ij}]$. The relative coordinate of each device in the multidimensional space is expressed as matrix $X_{n\times\omega}$, where n is the number of devices and ω is the

dimension of the relative coordinate of each device. The distance between the relative coordinates of each device is expressed by $d_{ij}(X)$. MDS takes advantage of device dissimilarity and iterates to make p_{ij} and $d_{ij}(X)$ as consistent as possible. The proximity between p_{ij} and $d_{ij}(X)$ is measured by the STRESS coefficient, and the specific calculation formula is shown as follows:

$$\text{STRESS} = \sum\left[f\left(p_{ij}\right) - d_{ij}(X)\right]. \tag{11}$$

Because there is no precise distance between devices in nonmetric multidimensional scales, the expression of nonmetric multidimensional scale is STRESS_1, replacing the dissimilarity p_{ij} in formula (11) with the value $\hat{d}_{ij}$ between devices, as shown in the following formula:

$$\text{STRESS}_1 = \sqrt{\frac{\sum_{ij,i\neq j}\left(\hat{d}_{ij} - d_{ij}\right)^2}{\sum_{ij,i\neq j}\left(d_{ij}\right)^2}}. \tag{12}$$

In both cases, nodes A and D cannot communicate directly due to the distance beyond their respective communication range. However, B and C are neighbors of A and D, and B and C can communicate directly. The Euclidean distance $L_{AC} = b$, $L_{C\,D} = d$, $L_{AB} = a$, $L_{B\,D} = c$, and $L_{BC} = p$ between each neighbor node can be obtained by TDOA ranging method. The required distance is the value q of $L_{A\,D}$. Let the angle between AB and BC be θ_1, and the angle between BC and BD be θ_2. According to the knowledge of plane trigonometry, according to the above two cases, equations as shown in formulas (13) and (14) can be obtained.

$$\begin{cases} b^2 = a^2 + p^2 - 2ap \times \cos\theta_1, \\ d^2 = c^2 + p^2 - 2cp \times \cos\theta_2, \\ q^2 = a^2 + c^2 - 2ac \times \cos(\theta_1 + \theta_2), \end{cases} \tag{13}$$

$$\begin{cases} b^2 = a^2 + p^2 - 2ap \times \cos\theta_1, \\ d^2 = c^2 + p^2 - 2cp \times \cos\theta_2, \\ q^2 = a^2 + c^2 - 2ac \times \cos(\theta_1 - \theta_2). \end{cases} \tag{14}$$

There are two sets of solutions that satisfy the equations, and the angle values of θ_1, θ_2, and q are also obtained. The position of D in the two sets of solutions is symmetric with respect to BC, that is, D and D'. However, there is only one actual position of D, so we need to use other external conditions to get the real position of D and then get the real Euclidean distance of AD.

In the above ideal case, the accurate distance between AD can be calculated with the presence of auxiliary equipment. However, due to the existence of hardware level and interference factors, the distance measurement between neighbor nodes will produce errors in the specific application. It is difficult for the quadrilateral composed of auxiliary equipment to get completely equal solutions due to the existence of distance measurement errors, so error correction of the above method is needed. The specific correction method is as follows: Firstly, the interference solution is eliminated by using the defined error function, and then the estimated AD distance is obtained by weighting multiple distance values with errors according to the node density and other factors.

Suppose there are several neighbor device pairs between nodes A and D that can form A quadrilateral with nodes A and D, and these qualified node pairs can form m quadrilateral that meets the above conditions. In this way, $2m$ distance solutions between nodes A and D are obtained according to the above method, including m estimated values with errors and m interference solutions. In order to eliminate the interference solution, the error function is defined as shown in the following formula:

$$e(u) = \sum_{i=1}^{m}\left(\min\left(\left|u - q_i\right|, \left|u - q_i'\right|\right)\right)^2. \tag{15}$$

According to the geometrical relation of m quadrilateral, the value range of q is between $(\min(q_1, \ldots, q_m, q_1', \ldots, q_m'), \max(q_1, \ldots, q_m, q_1', \ldots, q_m'))$. When the number of nodes in the cluster is not large, the fixed step method is adopted to reduce the amount of calculation and substitute into the above formula for search. The optimal value of q is q_{best}, which makes $e_q \leq e(u)$. Then, the m solutions closest to q_{best} are obtained from the m pairs of solutions, thus eliminating the m interference solutions. The next step is to get an estimate of q by weighting. In general, the higher the node density is, the greater the accuracy of positioning is. Therefore, the solution with the higher node density is set with a higher weight. The node density can be expressed by the device connection degree C.

$$q = \frac{\sum_{i=1}^{m} q_i \times (C_{ui} + C_{vi})}{(m-1) \times \sum_{i=1}^{m} C_i}. \tag{16}$$

The calculation formula for distance between devices after eliminating errors is as follows:

$$d = q \cdot u(u) - e(u). \tag{17}$$

In the above formula, $u(u)$ represents the initial calculation result of distance.

The positioning method of Internet of Things devices proposed in this paper includes network initialization, relative coordinate system establishment within cluster, and absolute coordinate transformation. The specific steps are as follows:

(1) The network starts to initialize. First, each device broadcasts data packets to detect the remaining energy information of its neighboring devices, taking into account the energy consumption of the RF amplifier circuit, the distance between devices, and the number of bits of data sent and received by devices. At the same time, the distance measurement of the neighbor device is calculated by TDOA method, and the distance measurement result is saved in the neighbor table.

(2) The cluster head is elected according to the remaining energy information of each neighbor

node. The node with the largest remaining energy is declared as the cluster head, while the other nodes with smaller remaining energy are divided into multiple clusters. The network initialization is completed.

(3) At the beginning of positioning, the distance between multihop nodes in the cluster is estimated by using the distance error correction algorithm above according to the shortest path composed of the distance between single-hop nodes in the cluster and the geometric relationship between nodes.

(4) Each cluster is based on the distance and relative coordinates between multihop nodes in the cluster.

(5) The relative coordinate system is fused from the two adjacent clusters with the most common nodes until all clusters are fused into the same relative coordinate system.

(6) Convert the relative coordinates of nodes in the network according to the location information of anchor nodes. The absolute coordinates of each node are obtained, and the positioning is finished, so as to realize the positioning of Internet of Things devices.

4. Simulation Experiment Design

Basic conditions Experimental environment parameters are set as shown in Table 1.

4.1. Verifying Key Update Performance. OPNET was founded in 1986 by two Ph.Ds from the Massachusetts Institute of Technology and commercialized in 1987. OPNET has won a wide range of customers in manufacturing, business, operators, defense, and research institutes for its precise simulation approach to network simulation and the accuracy of the results presented. In order to meet the needs of scientific research, the teaching and research office introduced this software in 2007, and it has been applied in project simulation.

This section uses OPNET to simulate SST scheme and analyzes the performance of the scheme in the actual network environment. Experiment sets up a one-to-many network model composed of a key server and multiple user terminals. By setting the number of user terminals, simulate the size of different group members and measure the time delay of the key server to update the group key and the amount of data sent when a single group member joins or quits.

The RSA algorithm, AES algorithm, MD5 algorithm, and d-h key negotiation algorithm used in ACE whiteboard modeling are derived from Crypto++ cryptography library. The length of public and private keys of RSA algorithm is 1024 bit, and the degree of all keys in the key tree is 128 bit. Add user-defined statistics: Re_keying Time and Server Traffic Sent, which, respectively, indicate the average delay of group key update and the number of messages sent by the group controller.

Table 1: Experimental environmental parameters.

Parameter	Description
Computer version and model	ASUS X550
Image net dataset version	3.0.3
CPU	Intel i7-9700K
Memory	1 GB
Operating system	Windows10
Hard disk capacity	120 GB
Data sampling frequency	Collect data every 2 seconds
The simulation software	Matlab 7.2

In the process of simulation, several simulation sequences are established according to the different number of group members at the initial moment. The number of group members increased exponentially from 32 to 8192, with a total of nine simulation sequences. The scalar statistics generated during the simulation process are collected and the simulation results are obtained. The time interval for each member to join (quit) follows the exponential distribution with the parameter of 100 (s), and the simulation time is 24 hours.

Reference [9] method and reference [10] method are used as experimental comparison methods, and the test results of key update time delay of different methods are shown in Figure 2.

As can be seen from Figure 2, when the number of group members is 32, the key update cost of [9] method, [10] method, and the method in this paper is the lowest, with time delay of 0.12 s, 0.71 s, and 2 s respectively. However, compared with the method in reference, the key update time delay of the method in this paper is lower, and the practical application effect is better.

Reference [9] method and reference [10] method are used as experimental comparison methods, and the test results of key update traffic of different methods are shown in Figure 3.

When the number of group members is 8192, the key update cost of [9] method, [10] method, and the method in this paper is the largest, and the communication message volume is 760 byte, 1830 byte, and 1900 byte, indicating that the data amount used in the key update process of this method is lower.

4.2. Verifying Device Positioning Performance. In this paper, the Internet of Things device fingerprint is run on the public data set to verify whether the Internet of Things device can be found. The data comes from Censys, an open source platform for collecting data packets in application protocols, including industrial control protocols, HTTP, FTP, and Telnet. Censys data set on August 15, 2020, is used in this paper, as shown in Table 2.

The HTTP protocol comes from port 80, and the data set has 514 G, including 66 million data packets. The FTP protocol comes from port 21 and contains 32 gigabytes of data, including 10 million data packets. Telnet protocol comes from port 23, data volume of 3 G, including 3 million data packets. The SSH protocol comes from port 22 and

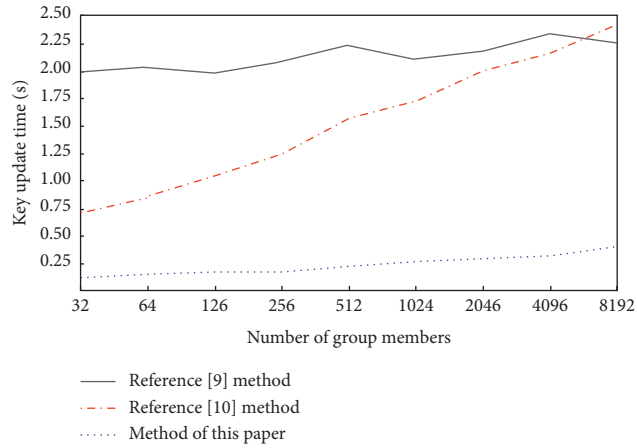

Figure 2: Key update time delay.

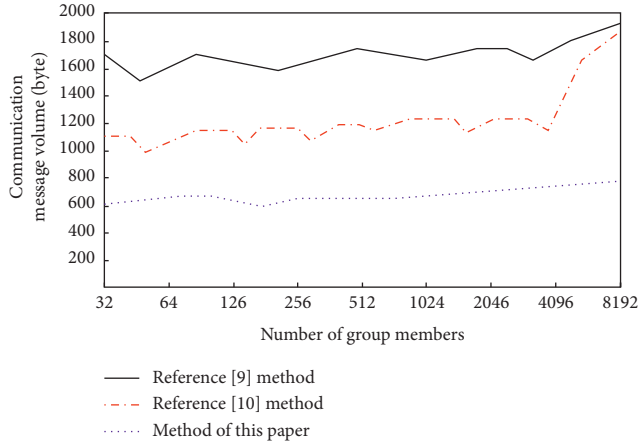

Figure 3: Communication data volume of key update.

Table 2: Source information of response packets.

The name of the protocol	The port number	Data volume size (G)	The time stamp
HTTP	80	512	2020-08-15
FTP	21	32	2020-08-15
Telnet	23	3	2020-08-15
ssh	22	50	2020-08-15

contains 50 gigabytes of data, including nearly 6 million data packets.

In this paper, the device fingerprints of routers, printers, and cameras are generated using a SVM classifier with 100 dimensional feature vectors and a radial basis kernel function. This paper found 5.52 million (552,333) Internet of Things devices, as shown in Figure 4.

That includes more than 3 million routers, 80,000 printers, and 2 million cameras. While the percentage of Internet of Things devices directly exposed to cyberspace is low, probably less than 1%, the absolute numbers are still huge. From a security perspective, these consumer-facing devices should not be visible or accessible from public spaces. These exposed Internet of Things devices could be exploited by malicious users and attackers. Many devices are hidden behind a private network (that is, a home or business network) and experiments have found that many devices return error codes indicating that they have no access.

Therefore, on the basis of the above, the positioning error rate and positioning time of Internet of Things devices in [11] method, [12] method, and this paper are compared, and the results are shown in Figures 5 and 6.

The analysis of the data in Figure 5 shows that, with the increase of the number of experiments, positioning errors of Internet of Things devices of different methods show a trend of change. The positioning error rate of Internet of Things devices of [11] method varies between −17% and 18%, and that of [12] method varies between −18% and 17%. The

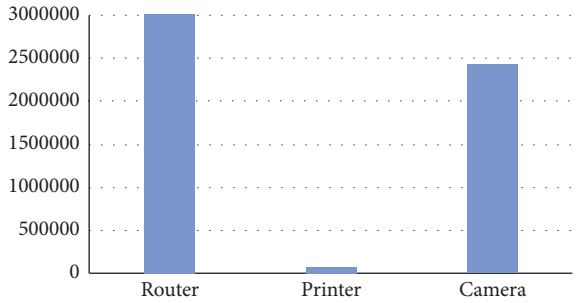

Figure 4: Device identification results.

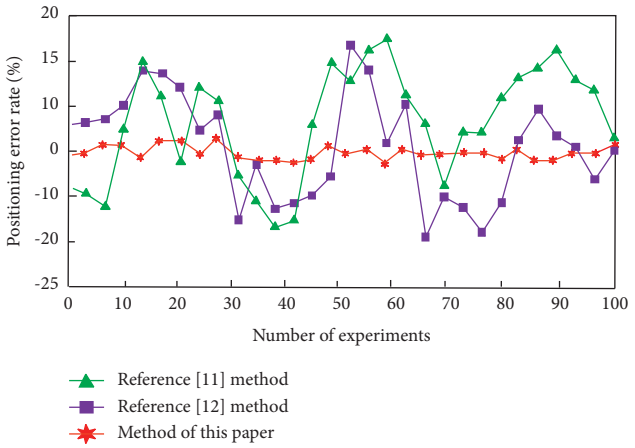

Figure 5: Comparison of positioning error rate.

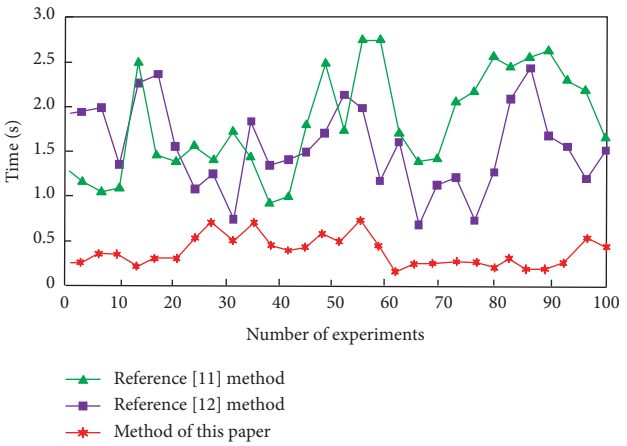

Figure 6: Comparison of positioning time.

positioning error rate of Internet of Things devices in this method varies between −4% and 3%. Compared with [11] method and [12] method, the positioning error rate of Internet of Things devices in this method is lower, indicating that the positioning accuracy of this method is higher.

By analyzing the data in Figure 6, it can be seen that the positioning time of Internet of Things devices in [11] method is between 0.8 s and 2.8 s, and that in [12] method is between 0.7 s and 2.5 s. Compared with [11] method and [12] method, the positioning time of Internet of Things devices in this method is less than 0.8 s, indicating that the positioning time of this method is shorter, which fully verifies the effectiveness of this method.

Experiments prove that this method has a lower key update time delay and a lower amount of data used in the key update process, and the positioning error rate of Internet of Things devices varies between −4% and 3%. The positioning time is less than 0.8 s. The positioning error rate of Internet of Things devices is lower and the positioning time is shorter, which fully verifies the effectiveness of this method.

5. Conclusion

With the development of Internet of Things technology, the number of intelligent devices increases rapidly. The self-organizing network mode adopted by wireless intelligent Internet of Things devices is characterized by dynamic topology, no central distribution, and no fixed infrastructure, etc. However, these characteristics bring many problems to the security of communication and positioning. Firstly, in order to achieve secure communication of wireless intelligent devices, often adopt the combination of single and double key encryption method; however, once a device key is lost or leaked, the key update in self-organizing network synchronization is in time, and the subsequent equipment positioning error rate is rising, so this paper proposes a new intelligent contract based on Internet of Things key updating and device positioning method. Experimental results show that the method has lower key update time delay and lower data amount used in the key update process, the positioning error rate of IoT devices varies between −4% and 3%, and the positioning time is less than 0.8 s, indicating that the method can be used to achieve accurate and rapid positioning of IoT devices, and the practical application effect is better. It can solve the problems existing in traditional methods and lay a solid theoretical foundation for the research on key update and device positioning of the Internet of Things.

Data Availability

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding this work.

Acknowledgments

This work was supported by Jiangxi Social Science “13th Five Year Plan” (2018) Project “Research on archives data management in the big data era” (no. 18TQ08).

References

[1] J. Li, D. Y. Jiang, W. J. Wang, and S. Y. Chen, “Electronic communication data link encryption simulation based on spatial sparse coding,” *Computer Simulation*, vol. 38, no. 8, pp. 190–193, 2021.

[2] J.-W. Hu, L.-Y. Yeh, S.-W. Liao, and C.-S. Yang, "Autonomous and malware-proof blockchain-based firmware update platform with efficient batch verification for Internet of Things devices," *Computers & Security*, vol. 86, no. 1, pp. 238–252, 2019.

[3] Y. Gu, H. Chen, Y. Zhou, Y. Li, and B. Vucetic, "Timely status update in internet of things monitoring systems: an age-energy tradeoff," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5324–5335, 2019.

[4] Y. Gui, S. M. Tamore, A. S. Siddiqui, and F. Saqib, "Key update countermeasure for correlation-based side-channel attacks," *Journal of Hardware and Systems Security*, vol. 4, no. 1, pp. 1–12, 2020.

[5] S. Medileh, A. Laouid, E. M. B. Nagoudi et al., "A flexible encryption technique for the internet of things environment," *Ad Hoc Networks*, vol. 106, no. 1, pp. 102240–102251, 2020.

[6] M. Ali, M. R. Sadeghi, and X. Liu, "Lightweight revocable hierarchical attribute-based encryption for Internet of Things," *IEEE Access*, vol. 8, no. 1, pp. 2169–3536, 2020.

[7] M. Al-Asli, M. E. S. Elrabaa, and M. Abu-Amara, "FPGA-based symmetric re-encryption scheme to secure data processing for cloud-integrated Internet of Things," *IEEE Internet of Things Journal*, vol. 23, no. 1, pp. 1–12, 2019.

[8] A. Diro, H. Reda, N. Chilamkurti, A. Mahmood, N. Zaman, and Y. Nam, "Lightweight authenticated-encryption scheme for internet of things based on publish-subscribe communication," *IEEE Access*, vol. 8, no. 1, pp. 60539–60551, 2020.

[9] W. Zhang, L. P. Tian, Y. Liang, and J. Deng, "Security authentication and key update method for multi-point cooperative joint transmission of Internet of vehicles," *Chinese Journal of Highway*, vol. 32, no. 6, pp. 308–318, 2019.

[10] Y. Y. Yao, X. L. Chang, and P. Zhen, "Decentralized identity authentication and key management scheme based on blockchain," *Cyberspace Security*, vol. 10, no. 6, pp. 33–39, 2019.

[11] Z. q. Huang, K. Wu, and G. W. Lu, "Global positioning method of high-value medical devices based on Internet of Things smart tags," *Chinese Journal of Medical Physics*, vol. 38, no. 9, pp. 1168–1171, 2021.

[12] Z. B. Ma and Q. Ren, "Application of blockchain technology in security positioning of Internet of Things equipment," *Automation technology and application*, vol. 39, no. 12, pp. 96–100, 2020.

[13] H. S. Bae, "The interaction effect of information systems of shipping and logistics firms and managers' support for blockchain technology on cooperation with shippers for sustainable value creation," *Sustainability*, vol. 13, no. 6, pp. 1–15, 2021.

[14] C. E. Ngubo and M. Dohler, "Wi-Fi-dependent consensus mechanism for constrained devices using blockchain technology," *IEEE Access*, vol. 8, no. 1, pp. 143595–143606, 2020.

[15] K. Ming, L. A. Yan, W. C. Chao, and E. Mltd, "A literature review of blockchain technology applications in supply chains: a comprehensive analysis of themes, methodologies and industries," *Computers & Industrial Engineering*, vol. 154, no. 1, pp. 107133–107146, 2021.

[16] X. Yang, T. Li, X. Pei, L. Wen, C. Wang, and E. Mltd, "Medical data sharing scheme based on attribute cryptosystem and blockchain technology," *IEEE Access*, vol. 8, no. 1, pp. 45468–45476, 2020.

[17] Z. Yang and H. Lei, "FEther: an extensible definitional interpreter for smart-contract verifications in Coq," *IEEE Access*, vol. 7, no. 1, pp. 37770–37791, 2019.

[18] T. Paul, "Electronic bills of lading, blockchains and smart contracts," *International Journal of Law & Information Technology*, vol. 23, no. 4, pp. 339–371, 2020.

[19] A. A. Sadawi, B. Madani, S. Saboor, M. Ndiaye, and G. Abu-Lebdeh, "A comprehensive hierarchical blockchain system for carbon emission trading utilizing blockchain of things and smart contract," *Technological Forecasting and Social Change*, vol. 173, no. 4, pp. 26–37, 2021.

[20] A. Kf, B. Zb, A. Ml, D. Avvc, and E. Ws, "Dredas: decentralized, reliable and efficient remote outsourced data auditing scheme with blockchain smart contract for industrial Internet of Things," *Future Generation Computer Systems*, vol. 110, no. 3, pp. 665–674, 2020.

[21] I. Ashraf, X. Ma, B. Jiang, and W. K. Chan, "GasFuzzer: fuzzing ethereum smart contract binaries to expose gas-oriented exception security vulnerabilities," *IEEE Access*, vol. 8, no. 1, pp. 99552–99564, 2020.

[22] J. An, "Framing regulation around the potential liabilities of parties in the blockchain & smart contract industry," *Fordham Journal of Corporate and Financial Law*, vol. 25, no. 7, pp. 1–11, 2020.

[23] A. Carvalho, "Bringing transparency and trustworthiness to loot boxes with blockchain and smart contracts," *Decision Support Systems*, vol. 144, no. 10, pp. 113508–113519, 2021.

[24] F. Peng, H. Tian, H. Quan, and J. Lu, "Data auditing for the internet of things environments leveraging smart contract," *Communications in Computer and Information Science*, vol. 21, no. 6, pp. 133–149, 2020.

[25] P. Kothari, A. Bharambe, R. Motwani, and A. Rathi, "Smart contract for real estate using blockchain," *SSRN Electronic Journal*, vol. 8, no. 1, pp. 1–16, 2020.

[26] X. D. Liang, X. Q. Deng, and H. E. Zi-Qi, "Smart contract and whole chain supervision:solutions to systematic barriers in cross-border trade of prescription drugs," *Journal of Changsha University of Science and Technology (Social Science)*, vol. 36, no. 7, pp. 25–34, 2019.

[27] M. Abraham, H. Aithal, and K. Mohan, "Real time smart contracts for internet of things using blockchain and collaborative intelligence based dynamic pricing for the next generation smart toll application," in *Proceedings of the 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pp. 206-207, Paris, France, 2020.

[28] X. Y. Wang, T. Y. Wu, C. Dong, S.-H. Zhao, and Y. Sun, "Prefixed-threshold real-time selection for correlated turbulent channel model for quantum key distribution with modulating retro-reflectors," *Quantum Information Processing*, vol. 20, no. 1, pp. 1–9, 2021.

[29] S.-H. Baek, S.-C. Yang, C.-Y. Park, C.-W. Park, S.-B. Cho, and S.-W. Ryu, "Room temperature quantum key distribution characteristics of low-noise InGaAs/InP single-photon avalanche diode," *Journal of the Korean Physical Society*, vol. 78, no. 6, pp. 634–641, 2021.

[30] Y. Huang, Y. C. Zhang, L. Huang, and S. Yu, "A modified practical homodyne detector model for continuous-variable quantum key distribution: detailed security analysis and improvement by the phase-sensitive amplifier," *ArXiv e-Prints*, vol. 54, no. 17, pp. 015503–015524, 2020.

[31] B. K. Park, K. W. Min, Y.-S. Kim, Y.-W. Cho, S. Moon, and S.-W. Han, "User-independent optical path length compensation scheme with sub-ns timing resolution for $1 \times N$ quantum key distribution network system," *Photonics Research*, vol. 8, no. 3, pp. 7–16, 2020.

[32] Y.-p. Yuan, C. Du, Q.-q. Shen et al., "Proof-of-principle demonstration of measurement-device-independent quantum key distribution based on intrinsically stable polarization-modulated units," *Optics Express*, vol. 28, no. 8, pp. 10772–10782, 2020.

[33] L. Chen, K. Huang, M. Manulis, and L. Chen, "Password-authenticated searchable encryption," *International Journal of Information Security*, vol. 21, no. 3, pp. 1–19, 2020.

[34] T. Bradley, J. Camenisch, S. Jarecki, A. Lehmann, G. Neven, and J. Xu, "Password-authenticated public-key encryption," *Applied Cryptography and Network Security*, vol. 29, no. 2, pp. 442–462, 2019.

[35] G. Demay, P. Gaži, U. Maurer, and B. Tackmann, "Per-session security: password-based cryptography revisited," *Journal of Computer Security*, vol. 27, no. 1, pp. 75–111, 2019.

[36] M. Park, G. Kim, Y. Park, I. Lee, and J. Kim, "Decrypting password-based encrypted backup data for Huawei smartphones," *Digital Investigation*, vol. 28, no. 12, pp. 119–125, 2019.