Blockchain-Based Privacy-Preserving Remote Data Integrity Checking Scheme for IoT Information Systems

Quanyu Zhao[a], Siyi Chen[a], Zheli Liu[b], Thar Baker[c], Yuan Zhang[a,*]

[a]*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China.*
[b]*College of Cyber Science, College of Computer Science, Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, China.*
[c]*Department of Computer Science, Liverpool John Moores University, United Kingdom, United Kingdom.*

Abstract

Remote data integrity checking is of great importance to the security of cloud-based information systems. Previous works generally assume a trusted third party to oversee the integrity of the outsourced data, which may be invalid in practice. In this paper, we utilize the blockchain to construct a novel privacy-preserving remote data integrity checking scheme for Internet of Things (IoT) information management systems without involving trusted third parties. Our scheme leverages the Lifted EC-ElGamal cryptosystem, bilinear pairing, and blockchain to support efficient public batch signature verifications and protect the security and data privacy of the IoT systems. The results of the experiment demonstrate the efficiency of our scheme.

Keywords: Data integrity checking, Privacy, Blockchain, Public verifiable, Security

1. Introduction

With the prolifieration of the Internet of Things (IoT) devices, including the including computers, cell phones, smart IoT equipments, etc., people are

*This is to indicate the corresponding author.
Email addresses: quanyu.zhao1105@gmail.com (Quanyu Zhao), yysfs1y1@gmail.com (Siyi Chen), liuzheli@nankai.edu.cn (Zheli Liu), t.baker@ljmu.ac.uk (Thar Baker), zhangyuan@nju.edu.cn (Yuan Zhang)

generating tremendous amounts of data every second. Storing and managing these data are bringing increasing challenge to ordinary Data Owners (DOs). Therefore, more and more DOs of IoT systems divert to cloud storage services provided by big companies such as Amazon, Google, etc, and run their information management systems in a remote manner.

Transferring data to the cloud servers is an effective method to reduce the storage pressure. However, the DO lose direct control of its data, which might lead to the security risks and privacy breaches. For instance, the cloud servers delete some low-inquiry or low-value data to reduce the storage cost, modify the primary data to cater the research departments'requirement. The individual's data always include sensitive or important information such as the medicine information, health information, the contract information etc.. Then, it will be a terrible disaster for DO if the data are sold, modified, destroyed or deleted intentionally or unintentionally. Some recent surveys [2–5] show that the integrity of the outsourced files is one of the major security concerns in the data storage management. Apparently, remote checking the integrity of the data is a popular method of guaranteeing both the privacy and the security.

1.1. Remote Data Integrity Checking

An excellent remote data checking scheme should have the following merits. Firstly, the most important ones are individual data security and privacy which guarantees that nobody can obtain the content of private data without permissions of DOs. If the individual data has modified or destroyed, this situation can always be found in the checking phase. Nextly, the operation right of the data should be controlled by the DO, and it should support dynamic data updating operations including modifications, insertions and deletions. In addition, the remote data integrity checking scheme should against the internal or external attacks, such as procrastinating auditors, malicious cloud servers, etc.. Last but not least, the efficiency of the integrity checking scheme is also a key factor for the practice application in the data management.

Uploading the huge amounts of individual's data to the cloud servers, the

DO will delete the local files to relieve the storage stress. With the limits of bandwidth, it is usually impossible to download the entirety of the data to check its integrity. Then, the auditors in the existing remote data integrity checking schemes always employ probabilistic verification to check the integrity of the data. The probabilistic verification method chooses a proportion of data for checking once. This probabilistic verification checking method guarantees the security, meanwhile, it satisfies the practice application requirements.

1.2. Related work

With the value of outsourced data becoming more and more important, the distributed and online storage systems are facing greater security and privacy challenge. Checking the integrity of the data is a method for protecting the security and privacy. The first remote data integrity checking was proposed by Blum et al in 1994 [2]. The process of checking the integrity of the data didn't reveal any information about the entire data. The cloud servers were assumed to be trustless in the Provable Data Possession (PDP) schemes [3, 4]. The DO divides the data into some metadata, stores the files and the metadata on the cloud servers, and deletes the local storage files. A small part of the metadata was chosen randomly to check the integrity of the data without the entire file and the summary of the data. Checking the correctness of the response from the storage server is equivalent to guaranteeing the unaltered files. This probabilistic verification method is suitable for checking operation and reducing computing and communication costs.

Another PDP scheme [6] supports dynamical updating their files after they stored their data on the cloud servers, the operation includes the updating, modification, deletion, and appending. It offered an efficient way to check the integrity of the data with a weakness of querying the data in limited times and not supporting the insertion operation. Erway et al [15] used Merkle Hash Tree (MHT) to improve the previous PDP schemes. However, Wang et al [11] point out that MHT tree cannot verify the indices of the data block, and it might suffer the replace attack. A few mechanisms [7, 8, 11–15] were proposed to check

the integrity of the outsourced data with the ability of the dynamical operation. Traditional cryptography techniques, such as message authentication codes and digital signatures, cannot be used to audit the integrity of the data. Then, integrity checking of remote data on the cloud servers is a big challenge and attracts more and more researchers. Some schemes [16–23] employed the cryptography techniques such as the public key and bilinear pairing, homomorphic encryption, digital signature to establish the mechanisms.

Batch verification can reduce the computational cost and the communication overhead of transmitting integrity tags. Shen et al [33] employs the batch verification to establish the secure real-time traffic data aggregation for vehicular cloud. Hu et al [34] proposes the autonomous and malware-proof blockchain-based firmware update platform, it is similar to other schemes [35, 36] which adopt the batch verification to increase the efficiency. We also employ this technology to enhance the efficiency of checking in our scheme.

Usually, there are always two participants in the data checking scheme, including DO and the cloud servers. DO checks the integrity of their data by carrying out a "two-parts" remote data checking protocol based on the fog computing [9, 10]. However, the audit result from either the DO or the cloud servers might be regarded as unreliable result for another participant. Then, a third party auditor with the access and the capacity of verifying the integrity of the outsourced data is employed to achieve the public verification in some protocols [11, 13]. However, the third party increases the privacy and the security threat in the auditing phase. They may reveal some information for some benefits in the absence of others, or be curious of others' information. The owners' private information or sensitive information is disclosed either actively or passively since the auditor may collude with the cloud servers or other attackers. Finding a trusted third party to audit the integrity of the data without revealing privacy is a big challenge in practice. Some previous works [3, 15, 24–26] have solved this problem. Two PDP schemes [3] based on the RSA number verify the integrity of the data without the third party, and both of them have a high computation cost. DPDP [15] proposed an efficient auditing scheme to verify the integrity of

the data with the ability of supporting the dynamic updating.

The situation that a fix auditor may be corrupted, is not conducive to the integrity of the data. The public verification is one of the most characteristics which means an external auditor or each entity with the ability of verifying the integrity of the data can play a role as the auditor in the auditing scheme. Ateniese et al proposed two PDP schemes [3] to achieve public verification firstly, and some works [8, 13, 16] based on their constructions provided some other characters in additional public verification.

The privacy of the outsourced data in the remote data integrity checking scheme [31] was defined by the requirements that the verifier cannot gain the entire blocks from the verification process. However, this definition of privacy isnnot widely accepted since each block may contain some confidential or sensitive data and it cannot be captured by others. Wang et al [32] presented the concept of zero knowledge public auditing, which means that the verifier gains nothing except the auditing information. In other words, the verifier attacks the audition data by guessing attack, and the probability of success is close to 0. Nevertheless, this work did not provide a formal security model.

Recently, a few schemes [1, 11–20, 24–26] were proposed to protect the privacy and security of the outsourced data by using a trusted third party. However, existing the trusted third party in the information system will increase the communication and computation complexity of the network. Under most conditions, it is difficult to find an always online trusted third party in practice, and it also brings the risks of privacy disclosure from the malicious third party auditor. Even through there is an always online trusted third party in practice. They may be single point of failure or suffer from the attack from the adversaries. This situation often happens in practice. Due to these shortcoming from the trusted third party, some works [3, 15, 25, 27–30] devote to remove the trusted third party from these schemes. However, these works mainly rely on complicated secure multiparty compution (SMC) techniques and are not suitable for IoT systems that involves devices with limited computation resources.

Considering blockchains' excellent advantages in terms of preventing data

loss and illegal modificaitons, we design a novel privacy-preserving remote data integrity checking scheme for IoT information systems based on the blockchain. The contributions of this paper are summarized as follows.

• In this paper, we propose blockchain-based privacy-preserving remote data integrity checking scheme without TTP, then we emphasize that our scheme is immune to privacy leakage from the third party and to collusion attacks of the cloud servers and the third party.

• Our scheme establishes a model based on the blockchain technology (a public distributed ledger). Once reaching the consensus, everyone, including the verifier, has the access to query the proof of the data for unlimited times from the blockchain, and cannot manipulate the data on the blockchain.

• Our scheme provides a solution to ensure the privacy-preserving public verification as well as the batch verification for multi-user data or multi-data simultaneously. It is secure against the malicious server and immune to the delayed auditor. Otherwise, it reduces the cost of the communication and storage.

• The security analysis and experiment results demonstrate that this protocol is suitable for a secure and practical real-world application.

The rest of the paper is organized as follows. Some preliminaries are presented in Section II. We put forward the system model in the Section III. Our construction and its analysis are proposed in the Section IV. Section V shows the performance evaluation of our scheme, and Section VI concludes our work.

2. Preliminaries

We utilize some techniques, including Lifted EC-ElGamal cryptosystem, bilinear pairing and aggregated signature, to establish blockchain-based privacy-preserving remote data integrity checking scheme without TTP for IoT. For the complete of our scheme, some preliminaries are detailed as follows.

2.1. Lifted EC-ElGamal cryptosystem

A few works [37–39] used the elliptic curve group to establish the Lifted EC-ElGamal cryptosystem. The Lifted EC-ElGamal cryptosystem is always

constituted by three components, including Key Generation, Encryption, Decryption.

(1) Key Generation. Use an elliptic curve group $E(F_q)$ with the order q, the generator G, the private and public key (X, Y) are generated by computing $Y = X \cdot G$.

(2) Encryption. The data $m \in L$, $L = \{0, 1, \cdots, t\}$, $(t \ll q)$, DO runs the encryption algorithm to obtain the ciphertext $C = (c, c^{'}) = (r \cdot G, m \cdot G + r \cdot Y)$, where r is a random number from Z_q^*.

(3) Decryption. The data m can be recovered using the Pollard's lambda algorithm [40] with the time complexity $O(\sqrt{t})$. The ciphertext was decrypted by solving the Equation 1.

$$m = \log_G(c^{'} - x \cdot c) \tag{1}$$

Due to the character of homomorphism in encrypted and decrypted algorithm, the encryption method and the ciphertext can be batch encrypted and decrypted by using the Equation 2 and Equation 3 respectively. Then, the DO downloads some ciphertexts and recovers the plaintext in the aggregated form or the individual form.

$$C_3 = C_1 + C_2 = (c_1 + c_2, c_1^{'} + c_2^{'}) = (r_3 \cdot G, (m_1 + m_2) \cdot G + r_3 \cdot Y) \tag{2}$$

$$m = m_1 + m_2 = \log_G[(c_1^{'} + c_2^{'}) - x \cdot (c_1 + c_2)] \tag{3}$$

2.2. Bilinear pairing

A bilinear pairing [41] maps two Gap Diffie-Hellman group G_1, G_2 elements to another multiplicative cyclic group G_3 element. Where g_1, g_2 are the generators of the G_1 and G_2 with the same order q. Therefore, three properties of the bilinear pairing have been listed as follows:

- Bilinear. $\forall a, b \in Z_P$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
- Non-degenerate. $e(g_1, g_2) \neq 1$.
- Efficient computation. $e(v_1, v_2)$ can be solved in the polynomial time for $\forall v_1 \in G_1$, $\forall v_2 \in G_2$.

2.3. Aggregated signature

The aggregated signature is a digital signature with the ability of converging some signatures into one signature and batch verifying its, such as CL-signature scheme [42]. The CL-signature scheme includes three algorithms described which were listed as follows.

(1) Key Generation. Based on an elliptic curve group $E(F_q)$ with the order q , the generator g_1, g_2, g_3, hash function $H(\cdot)$. User U_i, $(i = 1, 2, \cdots, n)$ generates the private and public key (x_i, y_i), satisfied $y_i = x_i \cdot g_1$.

(2) Signature. User U_i computes the signature σ_i by the Equation 4, where $h_i = H(m_1)$

$$\sigma_i = x_i \cdot g_2 + x_i \cdot h_i \cdot g_3 \tag{4}$$

(3) Batch verification. Using the signature σ_i and m , we verified the signature by solving the Equation 5.

$$e(\sum_{i=1}^{n} \sigma_i, g_1) = e(g_2, \sum_{i=1}^{n} Y_i) \cdot e(g_3, \sum_{i=1}^{n} H(M_i) \cdot Y_i) \tag{5}$$

3. System Model

We consider the data outsourced server that the DO has a large number of files beyond the ability of the local storage which should rent the storage space from the cloud. The cloud servers are equipped with enough storage space and computation resources. The cloud servers may destroy the individual data unintentionally or intentionally. The DO, the cloud servers or the external auditors who have expertise and capabilities of doing the verification work, can play the role of the auditor to check the integrity of the data for an unlimited number of times, and the auditor will receive some rewards for the verification work. Moreover, we take the privacy and security of the data into account. The architecture of the remote data checking auditing is illustrated in Figure 1, which includes four entities, including the DO, the cloud servers, the key generate center (KGC) and the auditors. The KGC runs the algorithm of generating the key for the entities, and publishes some public information.

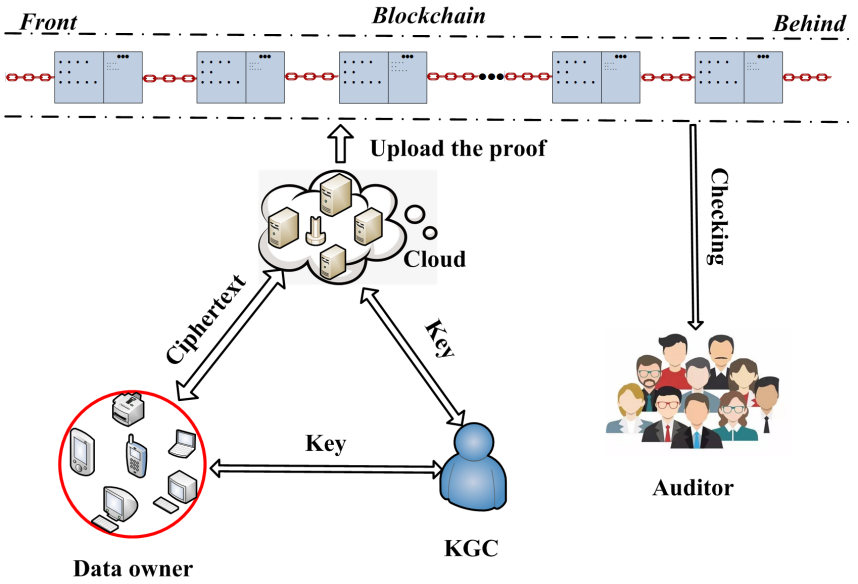

Figure 1: The model of the data checking

3.1. Trust Model

This work focuses on the outsourced data integrity checking in the cloud environment. In order to make the model more suitable for practical application in the information management system, some assumptions are put forward as follows.

The DO is semi-honest. The outsourced data was stored in the cloud servers, and the DO paid the fee for the storage. The DO may intend to reduce the fee of the data storage, and they may cheat the cloud servers in this method which they may pretend to be someone else and apply for the space storage.

The cloud servers are semi-honest. They are curious of the content of the outsourced data, and may sell the data to the other departments for some benefits. Moreover, they may increase the storage space of data for an additional fee. Last but not least, they always neglect the destroyed data, do not inform the DO to take the corresponding measures to reduce the damage and try their best to cheat the auditor or the DO. The cloud servers always generate a valid response for passing the verification without being detected when the data are

destroyed.

The auditor is a semi-honest entity, who is curious about the content storage data, and may leak the information of the individual data. They use all their resources and computation power to obtain some useful information. Using the additional resource and computation to audit the outsourced data is another way to gain some benefits.

3.2. Design Goals

To ensure the privacy and security of the outsourced data, we aim to propose a protocol for verifying the integrity of the data which achieves the following goals.

Correctness: The protocol ensures that the cloud servers cannot pass the audit if they modify or erase the data stored by DO.

Privacy: The auditor gains no more information than the integrity checking result.

Dynamic updating: The protocol should allow the DO to perform updating operations including the modifications, insertions, deletions and appending on its own data.

Public verification: The auditor, the DO, the cloud servers or an external auditor has the ability to check the integrity of the outsourced data.

Security: Operations including data modification, insertion, deletion, appending, and destroying can be performed only after being authorized by the DO.

4. Our Construction and Its Analysis

In this section, we put forward our construction primitively. Nextly, we show that our scheme achieves the properties of the correctness, security, privacy and dynamical updating.

4.1. Our Construction

Our construction consists of three phases, including setup, storage, and verification phase. The detail of each phase is described as follows.

Setup phase

Based on the elliptic curve group $E(F_q)$, the KGC gains the public parameter $\{E(F_q), q, g_1, g_2, Z_p^*, t\}$ and uploads it to the blockchain. The symbol q is the order of the group and g_1, g_2 are the generators of the group. Z_p^* represents an additive group and a set $A = \{0, 1, 2, \cdots, t\}$, which $t \ll q$.

The DO stores the large number of files on the cloud servers. Then, the DO and KGC executes the key generation algorithm in the Lifted EC-ElGamal cryptosystem to obtain the private key x_i and the corresponding public key Y_i by computing $Y_i = x_i \cdot g_1$. The cloud servers gain the private key and the corresponding public key (x_c, Y_c) by using this method.

Storage phase

In this phase, the DO divides those files into some data blocks, encrypts the blocks into the ciphertexts, signs it, and sends the ciphertexts and signatures to the cloud servers. The cloud servers verify the signatures and stores the signatures and ciphertexts. Thereafter, the cloud servers generate the abstracts of the ciphertexts and uploads the abstracts to the blockchain. It is detailed as follows.

Step 1. Due to the limited storage space, the DO rents the external space to store his data. Then he divides his data M into some data blocks m_j, $(1 \leq j \leq n)$ by using the RS code [43], and the blocks are named f_j, $(1 \leq j \leq n)$, denoted the set of the data blocks by $L = \{m_1, m_2, \cdots, m_n\}$.

Step 2. For each data block m_j, the DO chooses a random number $r_j \in Z_P^*$, and encrypts the data m_j into the ciphertext $C_i = (c_{i_1}, c_{i_2})$ by computing the matrix 6 and 7.

$$C_{i_1} = (c_{11}, c_{12}, \cdots, c_{1n}) = (r_1, r_2, \cdots, r_n) \begin{pmatrix} g_1 & & & \\ & g_1 & & \\ & & \ddots & \\ & & & g_1 \end{pmatrix} \tag{6}$$

$$C_{i_2} = (c_{21}, c_{22}, \cdots, c_{2n})$$
$$= (r_1, r_2, \cdots, r_n) \begin{pmatrix} Y_1 & & & \\ & Y_1 & & \\ & & \ddots & \\ & & & Y_1 \end{pmatrix} + (m_1, m_2, \cdots, m_n) \begin{pmatrix} g_1 & & & \\ & g_1 & & \\ & & \ddots & \\ & & & g_1 \end{pmatrix} \quad (7)$$

Step 3. When the ciphertext C_i is generated, the signature $Sig_i = (sig_{i1}, sig_{i2}, \cdots, sig_{in})$ of the data m_j, $(j = 1, 2, \cdots, n)$ is generated by the Equation 8.

$$Sig_{ij} = x_i H_1(ID_i||Y_i||fn_j||c_{1j}||c_{2j})g_2 \quad (8)$$

where ID_i is the identity of the DO, the file name is represented by fn_j, and the secure hash function is represented by $H_1(\cdot)$.

Step 4. After gaining the signature and the ciphertext of the data m_j, the DO computes and uploads the ciphertext $\{ID_i, Sig_{ij}, C_i, fn_j\}_{Y_c}$ to the cloud servers, and pays for the storage.

Step 5. In this step, the cloud servers verify the validity of the fee. Thereafter, the cloud servers verify the signatures of the files by checking the Equation 9.

$$e(Sig_{ij}, g_1) = e(g_2, H_1(ID_i||Y_i||fn_j||c_{1j}||c_{2j})Y_i) \quad (9)$$

Due to the homomorphic character of the signature algorithm, the signature can be batch verified by the following Equation 10.

$$e(\sum_{j=1}^{n} Sig_j, g_1) = e(g_2, \sum_{j=1}^{n} H_1(ID_i||Y_i||fn_j||c_{1j}||c_{2j})Y_i) \quad (10)$$

Step 6. The cloud servers obtain the tags τ_j of each ciphertext C_{ij}, $(1 \leq j \leq n)$ by the Equation 11.

$$\tau_j = x_c H_1(ID_i||Y_i||fn_j||c_{1j}||c_{2j}||r_{c,j})g_2 \quad (11)$$

Where $r_{c,j} \in Z_P^*$ is a random number. Then the cloud servers upload $\{ID_i, \tau_j, fn_j, r_{c,j}\}$ to the blockchain. Moreover, the cloud servers store $\{ID_i, \tau_j, C_i, fn_j, r_{c,j}\}$.

Step 7. The DO batch verifies the information $\{ID_i, \tau_j, fn_j, r_{c,j}\}$, $(1 \leq j \leq n)$

in Equation 12. The DO deletes the local blocks m_j, $(1 \leq j \leq n)$, if it is correctness.

$$e(\sum_{j=1}^{n} \tau_j, g_1) = e(g_2, Y_C \sum_{j=1}^{n} H_1(ID_i||Y_i||fn_j||c_{1j}||c_{2j}||r_{c,j})) \tag{12}$$

Verification phase

When the data owner downloads the data or desires to check the integrity of the individual data, they will publish the checking requirement on the taskbar. Then, the internal (include DO and the cloud servers) or external entities will regard as the auditor to launch the integrity verification of the files. The auditing process can be narrated as follows.

Step 1. The auditor queries the blocks from the cloud servers. Then the cloud servers send the ciphertext C_j, $(1 \leq j \leq n)$ to the auditor.

Step 2. The auditor queries the blockchain, obtains $\{ID_i, \tau_j, fn_j, r_{c,j}\}$, $(1 \leq j \leq n)$, and verifies the signatures of the blocks by the Equation 13.

$$e(\sum_{j=1}^{n} \tau_j, g_1) = e(g_2, Y_c \sum_{j=1}^{n} H_1(ID_i||Y_i||fn_j||c_{1j}||c_{2j}||r_{c,j})) \tag{13}$$

If the equation is correct, it means the cloud servers do not modify the blocks. Thereafter, the auditor uploads the checking transaction including the checking text, result, time and the auditor's signature to the blockchain.

4.2. Analysis

Therefore, we formally prove the correctness, security, privacy and discuss the dynamical analysis. The details of its analysis are described as follows.

Correctness

The DO rents the external space to store his/her data. The correctness of the data is very important for the DO since it may include some important or sensitive information such as contract information, healthy information, financial information and so on.

Lemma 1. The correctness of our scheme means the cloud servers' illegal operation can pass the checking with a negligible probabilistic.

Proof : The auditor downloads $\{ID_i, \tau_j, fn_j, r_{c,j}\}$ from the blockchain, and gains the ciphertext from the cloud servers. If the Equation 12 or the Equation 13 holds, the auditor believes the signatures and the information stored on the blockchain can be modified with a negligible probabilistic.

The cloud servers modify, destroy, delete, insert on the data without the DOs' permission, he will try the best to cheat the auditor for passing the checking. It means that the cloud servers should gain another random number $r_{j1} \in Z_P^*$ and compute the tags τ_{ij} by the Equation 14.

$$\tau_{ij} = x_c H_1(ID_i||Y_i||fn_j||c_{1j}||c_{2j}||r_{j,1})g_2 \tag{14}$$

If the tags $\tau_j = \tau_{ij}$ and $r_{j1} \neq r_j$ simultaneously, then the cloud servers pass the checking. However, it is infeasible for the cloud servers to find r_{j1} by using the PPT arithmetic such that $\tau_j = x_c H_1(ID_i||Y_i||fn_j||c_{1j}||c_{2j}||r_{j,1})g_2$. If the signature is changed, the illegality operation will be found with un-negligible probabilistic. Then, the cloud servers' illegal operation can pass the checking with a negligible probabilistic. This concludes the proof of the Lemma 1. □

In the proposed protocol, the integrity of the data can be checked. If the data was modified, destroyed, deleted and inserted, the tags cannot pass the checking process. Moreover, the checking transaction will be uploaded to the blockchain, which is a distribute ledge storing on all nodes. If the auditor cheats the DO in the checking process, the checking transaction will be tracked in the next verification phase.

Security

Before proving the security of the proposed protocol, we review the ECC encryption algorithm and the hash function in detail.

ECC encryption algorithm (Gen, Enc, Dec) is an asymmetric indistinguishable encryption algorithm with the presence of the attackers. For the whole probabilistic polynomial-time adversaries A and all i , there exists a negligible probabilistic ε such that

$$Pr[A(1^i, Dec_k(C_i)) = m] \leq \varepsilon \tag{15}$$

The adversaries acts as cloud servers, auditors or others external entities. The probability is taken over the adversaries A , the random choice of the data m and the key k , and any random choice in the decryption process.

Except the ECC encryption algorithm, we employed hash function and bilinear pairing to establish the proposed scheme. Hash function $y = H(x)$ is a one-way function with the character of gaining y from x easily. On the opposite side, it is infeasible to obtain x from y . Obtaining the fixed length of the output from any length of input is another character of the hash function. It is infeasible to find two different input messages with the same output.

Lemma 2: The proposed protocol is secure against the adversaries from internal or external with the polynomial-time attack algorithm.

Proof : As we all know, ECC encryption algorithm is a difficult problem that the adversaries have no polynomial-time attack algorithm to work on it. However, gaining the plaintext from the ciphertext in our scheme is equated with solving difficult discrete logarithms problem since the ECC encryption algorithm is one of the components of our scheme. Then, the plaintext cannot be obtained successful by the adversaries from internal or external with the equipment of the polynomial-time algorithm and good performance hardware.

On the other hand, the ciphertext is generated by using DO's public key and carrying out the ECC encryption algorithm. Then, it can't be decrypted unless using DO's private key or attack the ECC encryption algorithm successfully. In other word, nobody, except the DO, can gain the plaintext. The DO obtains the aggregation data or single data by computing the Equation 16.

$$\sum_{j=1}^{n} m_j = \log_{g_1}(c_j^{'} - x_i \cdot c_j) = \log_{g_1}(\sum_{j=1}^{n}(m_j \cdot g_1 + r_j \cdot Y_1 - x_i r_j g_1)) \qquad (16)$$

This concludes the proof of the Lemma 2. □

Privacy

The meaning of the privacy in this protocol includes two sides of the interpretations. One of the interpretations is that the cloud servers cannot obtain any information in the storing phase. The other one is that no auditor gains any information in the verification phase.

Lemma 3: Under the semi-honest model, no entity gains any information about the DO's data.

Proof : Primarily, the signature Sig_j comes from the identity DO, the file name, ciphertext and the private key. The cloud servers verify the legality of the signature by using the information $\{ID_i, Sig_j, fn_j, C_j\}$ with the tool of the bilinear pairing. Due to the peculiarity of the bilinear pairing, the data is stored in the cloud servers in the occultation method. The cloud servers do not gain any information from the signature in the verification phase. Then, the privacy of the data is guaranteed on the cloud servers side.

The auditor from the internal or external entities with the ability of the computing power and computing resources will check the integrity of the data. The auditor may be curious of the content and try to obtain it. In the verification phase, the auditor downloads $\{ID_i, \tau_j, fn_j, r_{c,j}\}$, $(1 \leq j \leq n)$ from the blockchain, gains the C_j, $(1 \leq j \leq n)$ from the cloud servers and checks the integrity of the data by the Equation 13. In the process of the verification phase, no one will gain information with respect to the plaintext from the auditing information since it is difficult to solve the discrete logarithms problem. The auditor cannot obtain information about the data. Then the privacy of the data will be protected on the auditor's side. This concludes the proof of the Lemma 3. □

Dynamical

In practice, the DO wants to dynamic inquire and operate on their data. In our scheme, the data can be operated as modification, insertion, and deletion without downloading the whole original files. The dynamic operation analysis are shown as follows.

Modification: In the cloud storage management system, the DO modifies the data frequently. The DO often replaces the primary blocks m_j with new data blocks $m_j^{'}$. The DO chooses a random number $r_j^{'} \in Z_P^*$ and computes the *ciphertext* $C_{ij}^{'}$ and signatures $\widetilde{sig_{ij}}$ by Equation 17 and Equation 18.

$$C_{ij}^{'} = (c_{i_1,j}, c_{i_2,j}^{'}) = (r_j^{'} \cdot g_1, m_j^{'} \cdot g_1 + r_j^{'} \cdot Y_i) \tag{17}$$

$$\widetilde{sig_j} = x_i H_1(ID_i||Y_i||fn_j^{'}||c_{i_1,j}||c_{i_2,j}^{'})g_2 \tag{18}$$

The DO computes and sends $\{ID_i, \widetilde{sig_j}, C_{ij}^{'}, fn_j^{'}\}_{Y_c}$ to the cloud servers. The cloud servers send $\{ID_i, \widetilde{\tau_j}, fn_j^{'}, r_{c,j}^{'}, fm\}$ to the *blockchain* and store the $\{ID_i, \widetilde{\tau_j}, C_j^{'}, fn_j^{'}, r_{c,j}^{'}\}$, the symbol fm includes the meaning of the modification of the original file and some other meaning. Thereafter, the cloud servers delete $\{ID_i, \tau_j, C_j, fn_j, r_{c,j}\}$ from the local database.

Data Insertion: The files are stored on the cloud servers in a certain order. The DO may insert a new file $m_j^{'}$ before the file m_j . Then, the DO runs the encryption algorithm and signature algorithm to obtain the $C_{j\sim1}^{'}$ and $\widetilde{sig_{j\sim1}}$.

$$\widetilde{sig_{j\sim1}} = x_i H_1(ID_i||Y_i||fn_{j\sim1}^{'}||c_{i_1j\sim1}||c_{i_2j\sim1})g_2 \tag{19}$$

The cloud servers verify the signature $\widetilde{sig_{j\sim1}}$, upload $\{ID_i, \widetilde{\tau_{j\sim1}}, fn_{j\sim1}^{'}, r_{c,j}^{'}\}$ to the blockchain, and insert $\{ID_i, \widetilde{\tau_{j\sim1}}, C_{j\sim1}^{'}, fn_{j\sim1}^{'}, r_{c,j}\}$ before the file $fn_j^{'}$ in the cloud servers.

Data deletion: When the DO sends the file m_j deletion requirement to the cloud servers. The cloud servers delete the file m_j and other information about them. Then, the cloud servers upload a deletion information about the file m_j to the blockchain. The auditor verifies the deletion information simultaneously.

5. Evaluation

In this section, we analyze the probability of the illegality behavior detection, perform the comparison with some existing schemes, and conduct an experiment to test our scheme's efficiency.

Due to limited computation power and resources, we cannot download all the blocks to verify the integrity of the files. Then, the auditor always chooses some blocks to verify the integrity of the data randomly. This is a probabilistic verification. Assume that the files are divided into ω blocks, and the blocks are stored in the cloud servers. The cloud servers modify, delete or destroy u

Table 1: The probabilistic of the illegality being detection

$\frac{u}{v}$	10	20	30	40	50	60	70	80	90
5	4.91%	9.63%	14.15%	18.50%	22.66%	26.66%	30.48%	34.15%	37.66%
10	9.60%	18.37%	26.36%	33.64%	40.27%	46.29%	51.77%	56.73%	61.23%
15	14.09%	26.30%	36.38%	46.03%	53.93%	60.74%	66.60%	71.63%	75.95%

blocks with active or passive operation, the auditor chooses v blocks to verify the integrity. The probabilistic of the illegality operation being found is η . Then, η can be described in Equation 20.

$$\eta = P_r[x \geq 1] = 1 - P_r[x < 0] = 1 - \frac{C_{\omega-u}^{v}}{C_{\omega}^{v}} \tag{20}$$

When the cloud server's has some illegality operation, $P_r[x \geq 1]$ represents the probability of catching the cloud server's some illegality operation, and $P_r[x < 0]$ represents the illegality operation haven't be grasped. C_{ω}^{v} and $C_{\omega-u}^{v}$ express the combination algorithm in the way of choosing v entities from ω entities or $\omega - u$ entities randomly. Assume that $\omega = 1000$, Table 1 shows the probabilistic of the illegality being grasped.

From the Table 1, the probability of successful error detection will increase with the increasing number of the modified blocks. Even though a small number of the blocks are modified, the illegality operation can be grasped with a high probability by choosing enough number of blocks for auditing. If the cloud servers modify, delete or destroy 1% of the blocks, the auditor will detect the cloud server's illegality behavior with a probability of more than 99% when he or she has chosen 368 blocks to check the integrity of the data. If more blocks are modified, deleted, destroyed, the blocks chosen to check the integrity of the data will be less than 368, and the probability will be greater than 99%. Thus, the cloud servers modify a small part of the file, and more verification blocks should be verified in order to make the probability in grasping the illegality operation non negligible. No matter how many blocks are modified, the probability of the illegality behavior being detected is significant under the batch verification.

Table 2: Comparison with some related works

Scheme	Cryptography	Complexity				Dynamic auditing	Third-party auditor
		Server computation	Auditor computation	Communication	Verifier storage		
PDP [3]	Public-key	O(1)	O(1)	O(1)	O(1)	No	No
DPDP [13]	Public-key	O(log n)	O(log n)	O(log n)	O(1)	Yes	No
PADD [9]	Public-key	O(log n)	O(log n)	O(log n)	O(1)	Yes	Yes
RDPC [21]	Symmetric-key	O(log n)	O(log n)	O(log n)	O(1)	No	No
ESAOD [22]	Symmetric-key	O(1)	O(1)	O(1)	O(1)	No	Yes
Our scheme	Asymmetric-key	O(1)	O(1)	O(1)	0	Yes	No

We compare our scheme with other schemes in this subsection. The comparison includes the encryption algorithm, the complexity, the dynamic supporting, and the needing of the third party auditor. The complexity contains the computation complexity, the communication complexity, and the verifier storage complexity. The computation complexity obtains the encryption cost at the DO side, the signature cost at the DO side, the tag generating cost at the cloud servers side, the verification cost at the cloud servers side and auditor side. The storage complexity means the storage cost at the auditor side. Notes that n represents the max number of the encryption file. Taking the cost of the computation, storage and communication into consideration, the comparison among our scheme with other schemes is showed in Table 2.

As shown in Table 2, the related schemes keeps the storage cost at the verifier side at a constant size, but our scheme transfers the storage to the blockchain. This storage method has three benefits: first, it reduces the cost of storage on the verifier's side; next, it reduces the probability of the abstract being destroyed; last, if the auditor cheats the DO in the verification phase, we track the transaction on the blockchain to grasp the illegal auditor. The third-party auditor is appointed in the ESAOD scheme, PADD scheme. However, there is no third-party auditor scheme in the PDP scheme, DPDP scheme, RDPC scheme and our scheme. The existence of the third-party auditor may attract more attacks from the third-party auditor's side. If the abstract of the data is destroyed, it's impossible to check the integrity of the data.

The cloud servers may operate on the DO's privacy data, the auditor should

Table 3: The performance of our scheme

Size of the data	Key Gen(ms)	Enc(ms)	Dec(ms)	Sig Gen(ms)	Sig Ver(ms)	Abs Gen(ms)	Abs Ver(ms)
1024bits	25.054	0.956	1.248	0.322	0.357	0.179	0.314
2048bits		1.784	2.031	0.278	0.439	0.176	0.311
4096bits		3.498	3.986	0.245	0.348	0.181	0.314
8192bits		6.684	7.552	0.245	0.421	0.163	0.320

check some tags within a short time. It is more advantageous to aggregate different DO's signatures for checking at one time. In our scheme, we aggregate the signatures from one data owner or from different data owners into a single signature, and check the aggregation signatures by carrying out the batch verification algorithm. This batch verification method reduces the cost in the verification phase.

To show the efficiency of our scheme, we conduct the integrity checking experiment for the outsourced data. We operate the experiment on personal computer with equipping Intel(R) Core(TM) i5-6267U CPU @ 2.90GHz 8G RAM. We complete the programming in Python 3.7.1. ECC encryption in our scheme is achieved by the ElGamal algorithm, and the size of the key is set as 32bits. The size of the data varies from 1024bits to 8192bits. Then the result of the experiment is recorded the average time for the Setup algorithm, the signature generation and the verification, the abstract generation in the storage phase, the abstract verification in the public verification 10000 times. The record is list in Table 3.

Table 3 shows that the average time for the Setup algorithm is correlative to the size of the key, but not correlative to the size of the data, for it only includes the time of generation of the public key and the private key. The variation of the average time is shown in Figure 2. The average time for the encryption algorithm and the decryption algorithm varies linearly with the increasing size of the data. We reflect the average time of the encryption, decryption, the signature generation, verification, the abstract generation and verification in Figure 3. The Figure 3 shows that the average time for the signature generation, signature verification, abstract generation and abstract audition is not correlative to the

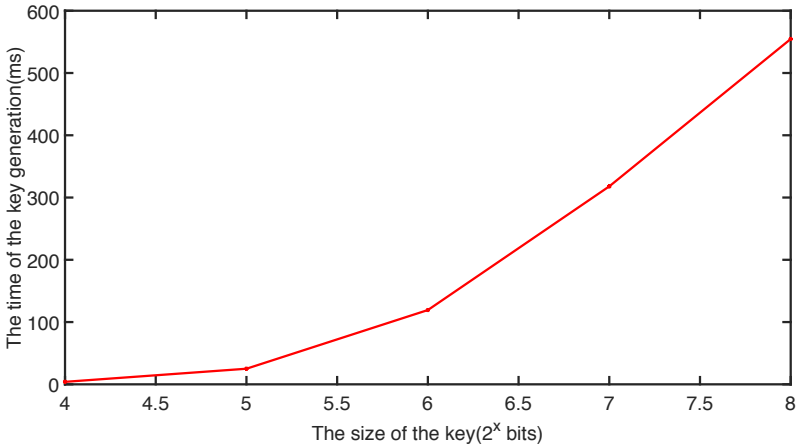

Figure 2: The variation of the average time with the size of the key

size of the data since those operations are on the ciphertext and the size of the ciphertext are correlative to the size of the data and the size of the key. The cost of the generated signature, signature verification, generated abstract and abstract verification increases with the size of the key varying from 16 bits to 256 bits. The result of the experiment shows that our scheme is efficient.

6. Conclusion and Future Work

In this paper, we propose a privacy-preserving remote data integrity checking scheme for IoT information systems without TTP based on blockchain. The proposed scheme is more suitable for practical applications in the data management system since it does not need a third party in the remote data integrity checking phase. Our scheme resists the leaking of data privacy caused by the un-trusted third party. We employ the blockchain technology to construct the model, then, the auditor queries the proof of the data for an unlimited number of times and uploads the auditing transaction to the blockchain. The DO or other auditors are able to track the illegal auditing transaction. Our scheme satisfies the correctness, privacy, dynamics, public verification and security. The results

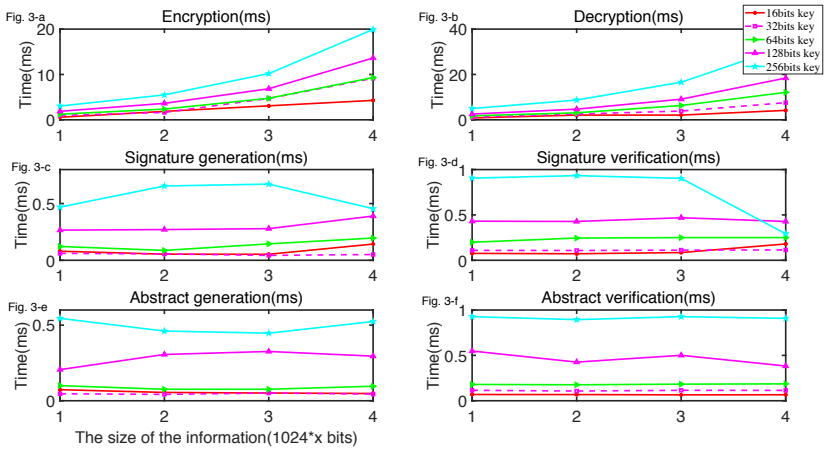

Figure 3: The average time of the cost. Fig 3-a. the average time of the encryption; Fig 3-b. the average time of the decryption; Fig 3-c. the average time of the signature generation; Fig 3-d. the average time of the signature verification; Fig 3-e. the average time of the abstract generation; Fig 3-f. the average time of the abstract verification; X axles are the size of the data in units of 1024 bits.

of the experiments demonstrate that our scheme is efficient in the communication and computation. We will research the automated remote data integrity checking in the future work. The automated remote data integrity checking means the integrity will be checked in real-time. If the data gets modified, deleted, destroyed or inserted illegally by the cloud servers or attackers, the DO will be notified in the first time. This would bring more time to the DO for dealing with the crisis.

7. Acknowledgements

This work was supported in part by National Key R&D Program of China (2018YFB1004301), NSFC-61872179, Fundamental Research Funds for the Central Universities (020214380052), NSFC-61425024, NSFC-61872176.

References

[1] Z. Liu, B. Li, J. Li.. NewMCOS: Towards a practical multi-cloud oblivious storage scheme. IEEE Transactions on Knowledge and Data Engineering. 2019. DOI:10.1109/TKDE.2019.2891581.

[2] M. Blum, W. Evans, P. Gemmell.. Checking the correctness of memories. Algorithmica. 1994. 12(2-3): 225-244. Doi.org/10.1007/BF01185212.

[3] G. Ateniese, R. Curtmola.. Provable data possession at untrusted stores. Proceedings of the 14th ACM Conference on Computer and Communications Security. Acm. 2007. 598-609. Doi.org/10.1145/1315245.1315318.

[4] G. Ateniese, R. Burns, R. Curtmola.. Remote data checking using provable data possession. ACM Transactions on Information and System Security. 2011. 14(1): 12. Doi.org/10.1145/1952982.1952994.

[5] O. Alfandi, S. Otoum, Y. Jararweh.. Blockchain solution for IoT-based critical infrastructures: byzantine fault tolerance. NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 2020, pp. 1-4. Doi:10.1109/NOMS47738.2020.9110312.

[6] G. Ateniese, R. Pietro, L. Mancini.. Scalable and efficient provable data possession. Preedings of the 4th International Conference on Security and privacy in Communication Netowrks. ACM. 2008. 9. Doi.org/10.1145/1460877.1460889.

[7] A. Barsoum, M. Hasan. Provable multicopy dynamic data possession in cloud computing systems. IEEE Transaction Information Forensics and Security. 2015. 10(3): 485-497. Doi:10.1109/TIFS.2014.2384391.

[8] Q. Wang, C. Wang, J. Li.. Enabling public verifiability and data dynamics for storage security in cloud computing. European Symposium on Research in Computer Security. Springer, Berlin, Heidelberg. 2009. 355-370. Doi:10.1016/S0031-8914(53)80099-6.

[9] Y. Huang, B. Li, Z. Liu.. Towards practical oblivious data access in fog computing environment. IEEE Transactions on Service Computing. 2019. Doi:10.1109/TSC.2019.2962110

[10] N. Tariq, M. Asim, F. Obeidat.. The security of big data in fog-enabled IoT applications including blockchain: A Survey. Sensors. 2019. 19(8): 59-69. Doi.org/10.3390/s19081788.

[11] Q. Wang, C. Wang, K. Ren.. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Transactions on Parallel and Distributed Systems. 2011. 22(5): 847-859. Doi:10.1109/TPDS.2010.183.

[12] K. Yang, X. Jia. An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Transactions on Parallel and Distributed Systems. 2013. 24(9): 1717-1726. Doi:10.1109/TPDS.2012.278

[13] Y. Zhu, G. Ahn, H. Hu.. Dynamic audit servers for outsourced storages in clouds. IEEE Transactions on Servers Computing. 2013. 6(2): 227-238. Doi:10.1109/TSC.2011.51.

[14] Y. Yu, Y. Li, J. Ni.. Comments on "public integrity auditing for dynamic data sharing with multiuser modification". IEEE Transactions on Information Forensics and Security. 2016. 11(3): 658-659. Doi:10.1109/TIFS.2015.2501728.

[15] C. Erway, A. Kp?, C. Papamanthou.. Dynamic provable data possession. ACM Transactions on Information and System Security. 2015. 17(4): 15. Doi.org/10.1145/2699909.

[16] J. Li, L. Zhang, J. Liu.. Privacy-preserving public auditing protocol for low-performance end devices in cloud. IEEE Transactions on Information Forensics and Security. 2016. 11(11): 2572-2583. Doi:10.1109/TIFS.2016.2587242.

[17] H. Wang, D. He, S. Tang. Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud. IEEE Transactions on Information Forensics and Security. 2016. 11(6): 1165-1176. Doi:10.1109/TIFS.2016.2520886.

[18] Y. Yu, M. Au, G. Ateniese.. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. IEEE Transactions on Information Forensics and Security. 2017. 12(4): 767-778. Doi:10.1109/TIFS.2016.2615853.

[19] G. Ateniese, S. Kamara, J. Katz.. Proofs of storage from homomorphic identification protocols. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg. 2009. 319-333. Doi.org/10.1007/978-3-642-10366-7_19.

[20] D. Boneh, B. Lynn, H. Shacham.. Short signatures from the Weil pairing. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg. 2001. 514-532. Doi.org/10.1007/3-540-45682-1_30.

[21] Z. Zhou. Abductive learning: towards bridging machine learning and logical reasoning. Science China Information Sciences. 2019. 062(007): 220-222. Doi.10.1007/s11432-018-9801-4.

[22] J. Li, Y. Huang, Z. Liu.. Searchable symmetric encryption with forward search privacy. IEEE Transactions on Dependable and Secure Computing. 2019. DOI:10.1109/TDSC.2019.2894411.

[23] K. Umair, M. Asim, B. Thar.. A decentralized lightweight blockchain-based authentication mechanism for IoT systems. Cluster Computing. 2020. 64(1): 59-69. Doi:10.1007/s10586-020-03058-6.

[24] T. Ristenpart, E. Tromer, H. Shacham.. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. Proceedings of the 16th ACM Conference on Computer and Communications Security. ACM. 2009. 199-212. Doi.org/10.1145/1653662.1653687.

[25] Y. Yu, Y. Zhang, J. Ni.. Remote data possession checking with enhanced security for cloud storage. Future Generation Computer Systems. 2015. 52: 77-85. Doi.org/10.1016/j.future.2014.10.006.

[26] Q. Gan, X. Wang, X. Fang. Efficient and secure auditing scheme for outsourced big data with dynamicity in cloud. Science China Information Sciences. 2018. 61(12): 122-104. Doi.org/10.1007/s11432-017-9410-9.

[27] J. Hua, A. Tang, Q. Pan.. Practical –anonymization for collaborative data publishing without trusted third party. Security and Communication Networks. 2017. 27: 1-10. Doi.org/10.1155/2017/9532163.

[28] Y. Liu, Y. Wang, X. Wang.. Privacy-preserving raw data collection without a trusted authority for IoT. Computer Networks. 2019. 148: 340-348. Doi:10.1016/j.comnet.2018.11.028.

[29] T. Butt, A. Iqbal, R. Salah.. Privacy management in social internet of vehicles: review, challenges and blockchain based solutions. IEEE Access. 2019. 7: 79694-79713. Doi:10.1109/ACCESS.2019.2922236.

[30] L. Tseng, L. Wong, S. Otoum.. Blockchain for managing heterogeneous internet of things: A perspective architecture.IEEE Network. 2020. 34(1). 16-23. Doi:10.1109/MNET.001.1900103.

[31] C. Wang, Q. Wang, K. Ren.. Privacy-preserving public auditing for data storage security in cloud computing. Infocom 2010. 2010. 1-9. Doi:10.1109/INFCOM.2010.5462173.

[32] C. Wang, S. Chow, Q. Wang.. Privacy-preserving public auditing for secure cloud storage. IEEE Transactions on Computers. 2013. 62(2): 362-375. Doi:10.1109/TC.2011.245.

[33] J. Shen, D. Liu, X. Chen.. Secure real-time traffic data aggregation with batch verification for vehicular cloud in VANETs. IEEE Transactions on Vehicular Technology. 2019. 69: 807-817. Doi:10.1109/TVT.2019.2946935.

[34] J. Hu, L. Yeh, S. Liao.. Autonomous and malware-proof blockchain-based firmware update platform with efficient batch verification for Internet of things devices. Computers and Security. 2019. 86: 238-252. Doi.org/10.1016/j.cose.2019.06.008.

[35] H. Shen, M. Zhang, J. Shen. Efficient privacy-preserving cube-data aggregation scheme for smart grids. IEEE Transactions on Information Forensics and Security. 2017. 12(6): 1369-1381. Doi:10.1109/TIFS.2017.2656475.

[36] J. Cui, J. Zhang, S. Zhong.. SPACF: A secure privacy-preserving authentication scheme for VANET with cuckoo filter. IEEE Transactions on Vehicular Technology. 2017. 66(11): 10283-10295. Doi:10.1109/TVT.2017.2718101.

[37] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory. 1985. 31(4): 469-472. Doi:10.1109/TIT.1985.1057074.

[38] Y. Desmedt. Threshold cryptography. European Transactions on Telecommunications. 1994. 5(4): 449-458. Doi.org/10.1002/ett.4460050407.

[39] Y. Liu, W. Guo, C. Fan.. A practical privacy-preserving data aggregation (3PDA) scheme for smart grid. IEEE Transactions on Industrial Informatics. 2018. 15(3): 1767-1774. Doi:10.1109/TII.2018.2809672.

[40] M. Grissa, A. Yavuz, B. Hamdaoui.. Preserving the location privacy of secondary users in cooperative spectrum sensing. IEEE Transactions on Information Forensics and Security. 2017. 12(2): 418-431. Doi:10.1109/TIFS.2016.2622000.

[41] S. James, N. Gayathri, P. Reddy.. Pairing free identity-based blind signature scheme with message recovery. Cryptography. 2018. 2(4): 29. Doi.org/10.3390/cryptography2040029.

[42] J. Camenisch, A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. Annual International Cryptology Conference. Springer, Berlin, Heidelberg. 2004. 56-72. Doi.org/10.1007/978-3-540-28628-8_4.

[43] J. Oh, J. Ha, H. Park.. RS-LDPC concatenated coding for the modern tape storage channel. IEEE Transactions on Communications. 2016. 64(1): 59-69. Doi:10.1109/TCOMM.2015.2504362.