IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

SPECIAL SECTION ON SECURITY AND PRIVACY IN EMERGING DECENTRALIZED COMMUNICATION ENVIRONMENTS

Received May 30, 2019, accepted June 17, 2019, date of publication June 24, 2019, date of current version July 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924486

A Secure and Efficient Data Integrity Verification Scheme for Cloud-IoT Based on Short Signature

HONGLIANG ZHU[1,2], YING YUAN[1,2], YULING CHEN[3], YAXING ZHA[4], WANYING XI[1], BIN JIA[5], AND YANG XIN[1,2]

[1] Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] National Engineering Laboratory for Disaster Backup and Recovery, Beijing 100876, China
[3] State Key Laboratory of Public Big Data, Guizhou University, Guizhou 550025, China
[4] Research and Development Center of Transport Industry of Network Security Technologies, China Communication Information Center Company Ltd., Beijing 100088, China
[5] Shandong University of Science and Technology, Qingdao 266590, China

Corresponding author: Hongliang Zhu (zhuhongliang@bupt.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2017YFB0802300, in part by the Major Scientific and Technological Special Project of Guizhou Province under Grant 20183001, in part by the Foundation of Guizhou Provincial Key Laboratory of Public Big Data under Grant 2018BDKFJJ008 and Grant 2018BDKFJJ020, in part by the NSFC under Grant U1836205, and in part by the Applied Sci-Tech R&D Special Fund Program of Guangdong Province under Grant 2015B010131007.

ABSTRACT The Internet of Things (IoT) is also known as the Internet of everything. As an important part of the new generation of intelligent information technology, the IoT has attracted the attention both of researchers and engineers all over the world. Considering the limited capacity of smart products, the IoT mainly uses cloud computing to expand computing and storage resources. The massive data collected by the sensor are stored in the cloud storage server, also the cloud vulnerability will directly threaten the security and reliability of the IoT. In order to ensure data integrity and availability in the cloud and IoT storage system, users need to verify the integrity of remote data. However, the existing remote data integrity verification schemes are mostly based on the RSA and BLS signature mechanisms. The RSA-based scheme has too much computational overhead. The BLS signature-based scheme needs to adopt a specific hash function, and the batch signature efficiency in the big data environment is low. Therefore, for the computational overhead and signature efficiency issues of these two signature mechanisms, we propose a scheme of data integrity verification based on a short signature algorithm (ZSS signature), which supports privacy protection and public auditing by introducing a trusted third party (TPA). The computational overhead is effectively reduced by reducing hash function overhead in the signature process. Under the assumption of CDH difficult problem, it can resist adaptive chosen-message attacks. The analysis shows that the scheme has a higher efficiency and safety.

INDEX TERMS Internet of Things, cloud computing, provable data integrity, privacy preserving, public auditability, short signature, ZSS signature.

I. INTRODUCTION

With the introduction and widely used of new concepts and technologies such as mobile Internet, intelligent transportation and smart city, the number of devices connected to the Internet is increasing and more powerful storage and processing resources are needed. The integration of cloud and Internet of things becomes an inevitable choice [1]. The Internet of Things connects all entities such as computers, mobile devices, and wearable smart devices to the Internet, giving them digital identities, then realizing communication and information sharing between objects. The cloud computing is introduced to store and process the huge amount of data collected by the IoT. Under this new network scenario, we can use network devices to remotely monitor and control any physical entity, makes reasonable decisions by embedding communication and computing resources in physical devices. The integration of cloud and Internet of things has greatly improved people's lives and work efficiency, it has been favored by more and more people. At the same time,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

the storage security of cloud and IoT is becoming particularly important [1], [2]. At present, integration of cloud and Internet of things is widely used in transportation systems, military, industrial manufacturing, medical care, smart home and other fields. However, as a new technology, it is not well secure. There exists a lot of security challenges in the proposed architecture such as data storage and privacy protection [3], [4].

The Internet of Things has limited storage and computing resources. However, the cloud can conveniently provide scalable storage resources and powerful computing resources. In recent years, IoT has expanded its capabilities by leveraging cloud resources in different ways. IoT stores data in the storage resources provided by cloud service providers (CSP). Also, the Internet of Things uses cloud computing to analyze, verify and store data, which greatly reduces the computation, storage and communication overhead of IoT and improves efficiency. To some extent, it meets the real-time requirements of IoT. However, as the storage resources provided by the CSP are relatively centralized, events such as hardware and software failures and malicious system damage in the storage system seriously threaten the secure storage of data. Under these circumstances, even if the data is lost due to the damage of the cloud storage system, devices in the Internet of Things are difficult to detect in time. Therefore, the data integrity checking is required in the IoT storage platform to ensure data integrity and availability [5]. Moreover, with the increase of data scale collected by sensors, how to efficiently carry out a data integrity verification (provable data possession, PDP) in a cloud storage server, reduce the computational overhead and communication overhead [6] of the storage server become a big challenge in IoT's storage security.

In order to solve the above problem, and improve the safety of IoT, we present a new model for data integrity verification based on a short signature, that is, ZSS signature [7] in this paper. The scheme can be roughly summarized as the following:

- Our scheme supports public auditing of user data by introducing a trusted third party (TPA). That is, users do not have to incur additional overheads for undertaking data.
- The scheme uses the random masking technique to preserving data privacy.
- In the signature process, we reduce the computational overhead of the hash function. The experimental results show that the computational cost of this scheme is smaller than that based on BLS.
- Under the assumption of the CDH problem, the scheme can resist adaptive chosen-message attacks and has high security in the random oracle model.

The remainder of this paper is structured as follows. We first review the related work in Section II. Next, Section III lists the basic concepts used in the ZSS signature mechanism. Section IV describes the basic definition and specific details of our data integrity verification scheme. In Section V, we analyze our scheme and experimental results. Finally, Section VI presents the conclusions and future work of the study.

II. RELATED WORK

Currently, a large number of IoT applications choose to store and process data on the cloud. The integrated network scenario of cloud and IoT is widely used. Neagu et, al provided a Cloud-IoT architecture (HM-SS) for health monitoring. This architecture leverages the inherent advantages of cloud computing such as scalable data storage, efficient processing resources, and controlled disaster recovery to improve the usage efficiency of sensor data. This system can help medical institutions monitor patients' conditions and provide remote guidance at any time [9]. With the widespread use of the cloud and IoT, the safety of integration is valued. Liu et al. [6] believe that most IoT applications choose to use the cloud to store and process data, and the cloud itself is not secure, so ensuring data integrity for the cloud-based Internet of Things (IoT) applications is a challenge. They proposed a blockchain-based data integrity service framework called DIaaS (Data Integrity as a Service), which enables data integrity verification in a fully decentralized environment. Literature [8] suggests that the integration of the Internet of Things and cloud computing is not entirely beneficial. With the ubiquitous computing that we will have in the future, data security and privacy will become a bigger issue. However, how to efficiently ensure the safety of the data. Unsurprisingly, a number of data integrity checking schemes have been proposed in the last few years. For data stored in the cloud server of the IoT storage system, the ''provable data possession'' (PDP) scheme can effectively verify the integrity of remote data to ensure storage security [10].

Regarding data integrity verification, Shah et al. [11] originally proposed a message authentication code-based PDP mechanism to verify the integrity of remote data by using a message authentication code as authentication metadata. However, such mechanisms only support a limited number of verifications and users are required to store a large amount of verification information. Storage overhead and communication overhead are large. For this reason, Venkatesh et al. [12], [13] consider constructing the PDP mechanism with RSA signature mechanism to check the integrity of remote data. Although this schema supports continuously infinite verification, it is computationally expensive for large file operations.

Yu et al. [14] proposed an identity-based cloud data integrity detection scheme, which solves the complex problem of traditional PDP certificate management and uses RSA signature to support public audit and data dynamics. However, the solution proved to be a security issue and vulnerable to data recovery attacks. Subsequently, Xu et al. [15] proposed a new identity-based public auditing scheme based on RSA signature for Yu's scheme.

For batch integrity verification of multiple copies of data, Liu et al. [16] proposed a multi-replica PDP scheme based on multi-replica Merkle hash tree, all replica blocks for each data block are organized into a same replica sub- tree. It supports full dynamic data updates, authentication of block indices and verification of updates for multiple replicas at the same time. Rajendran et al. [17] proposed identity-based PDP implementation using IBS scheme is suitable for integrity detection of data in multiple cloud storage. Liu et al. [16] proposed MUR-DPA audit scheme, which adopts the authenticated data structure (ADS) based on the Merkle hash tree (MHT) to realize dynamic data update, public audit and integrity authentication for multiple copies. Chen [18] constructs an algebraic signature-based remote data availability verification protocol with high efficiency and supports infinite verification. However, this protocol requires frequent challenge-update checks to achieve unlimited authentication, resulting in additional communication and computational overhead for the user. Fu et al. [19] proved that the protocol proposed by Chen is vulnerable to replay attacks by malicious cloud servers, which may lead to the loss of user data and leakage of private information. So, a new protocol was proposed [20]. It is based on algebraic signatures for data possession verification, allowing the third party to audit the integrity of outsourced data. It supports an unlimited number of integrity verifications and data privacy protections, but does not support data dynamic operations.

Wang et al. [21] proposed a scheme for provable data possession based on BLS homomorphic signature [22] and RS error-correcting codes. This scheme uses the Merkle hash tree and BLS signature mechanism to ensure the correctness of data blocks, supports public validation and data dynamics. Nevertheless, there may be a risk of leaking user privacy. Subsequently, a scheme for provable data possession was proposed. It uses random masking techniques to ensure data privacy and supports publicly audited [23]. Based on this scheme, Mukundan et al. [24] proposed a BLS-based data possession certification model that uses homomorphic tag technology to support public verification in a multi-copy cooperative storage environment.

Literature [25] proposes to integrate cloud computing and Internet-of-Things with physical medical equipment into a distributed network to build a health care system. A privacy-protected data integrity verification model is proposed by using a lightweight stream authentication data structure for the system. The model uses FHMT to ensure data integrity while using a symmetric encryption scheme to protect the privacy of user data. However, the program did not implement public verification.

In summary, most of the existing data integrity verification schemes are based on RSA [11]–[15] and BLS signatures [16], [21], [24]. The computational overhead is too heavy during the signing process based on the RSA scheme. The scheme based on the BLS signature requires a special hash function H:$\{0,1\}^* \rightarrow G_1$ to be used for signing the data, and the efficiency is low. In the bulk audit environment of the IoT, signature efficiency in the provable data possession process remains to be improved, and privacy protection needs to be further strengthened. Recently, Rossi and Schmid [26] proposed an identity authentication signature scheme based on ZSS, but their scheme is insecure and cannot resist forgery attacks and key disclosure attacks [27]. Therefore, this paper proposes a data integrity verification scheme based on ZSS short-signature [7] to improve the efficiency of the signature, reduce the computation and storage overhead in the signing process. At the same time, the scheme proved to be secure and could implement privacy protection and public auditing.

III. PRELIMINARIES

A. BASIC CONCEPTS

Definition 1 (Bilinear Mapping): Let G_1 and G_2 are cyclic groups of order q, and the generator of G_1 is P. Let $e : G_1 \times G_1 \rightarrow G_2$, be a map with the following properties [28]:

1) **Bilinearity:**

$$e(u^a, v^b) = e(u, v)^{ab},$$
$$e(u, kv) = e(ku, v) = e(u, k)e(u, v),$$
$$\text{for } \forall u, v, k \in G_1, \forall a, b \in Z_q.$$

2) **Computability:** For any $u, v \in G_1$, $e(u, v)$ is computable in polynomial time.

3) **Non-degeneracy:** There exists $\sigma \in G_1$ such that $e(\sigma, \sigma) \neq 1$, then, the e is non-degeneracy.

Definition 2 (Computational Diffie-Hellman Problem): Assuming that G_0, G_1 and G_2 are cyclic groups of order prime q, Let $e : G_0 \times G_1 \rightarrow G_2$, be a map with exists $h \in G_0, H \in G_1$, if $e(h, H) \neq 1$, then the e is non-degeneracy. Computing the probability $AdvCDH_A$ that the adversary A will solve the CDH difficulty problem:

$$\begin{aligned} AdvCDH_A &= \Pr[A(P, H, H^a) \\ &= P^a \,|\, P \overset{R}{\leftarrow} G_0;\, H \overset{R}{\leftarrow} G_1;\, a \overset{R}{\leftarrow} Z_{|G_1|}] < \varepsilon \quad (1) \end{aligned}$$

If adversary $\mathcal{A}$ can solve the above problem with non-negligible probability ε for all the polynomial time t, then the CDH problem is a (t, ε)- difficult problem. In other words, the mapping e is (t, ε)-secure, if and only if the CDH problem is (t, ε)- difficult problem.

Definition 3 (Security Signature): The forger acquires multiple message signatures on the polynomial from the message signed by the signer and has only the public key. In this situation, if it is not feasible to generate a correct message signature pair, then the signature mechanism S has unforgeability against adaptive chosen message attacks. In other words, for every probabilistic polynomial time forger algorithm $\mathcal{F}$ there does not exist a non-negligible probability ε [29].

Definition 4: After at most q_H hash prediction queries and q_S signature queries, if no polynomial bounded adversary $\mathcal{A}$ can output a correct forged signature with at least probability ε in time t, that is, no adversary can win the game (Challenge Games) with the advantage at least ε. So, signature scheme S is $(t, q_H, q_S, \varepsilon)$-secure under adaptive chosen message attacks, and it is unforgeable.

Definition 5 k -CAA(Collusion Attack With k Traitors [30]): This problem was proposed by Mitsunari et, al. in document [31] for the security basis of the traitor tracing scheme. In this article, we use it to prove the security of the signature mechanism ZSS. This algorithm means that let k be an integer, and $x \in Z_q,$, $P \in G_1$, given $\{P, Q = xP, h_1, \ldots., h_k \in Z_q, \frac{1}{h_1+x}P, \ldots, \frac{1}{h_k+x}P\}$, to compute $\frac{1}{h+x}P$ for any $h \in \{h_1, \ldots., h_k\}$. If for all polynomial time t, adversaries A have:

$$Advk - CAA_A = \Pr\left[\begin{array}{l} A(P, Q{=}xP, \dfrac{1}{h_1+x}P, \ldots, \dfrac{1}{h_k+x}P) \\ = \dfrac{1}{h_1+x}P \\ |x \in Z_q, P \in G_1, h_1, \ldots, h_k \in Z_q, \\ h \notin \{h_1, \ldots, h_k\} \end{array}\right] < \varepsilon \quad (2)$$

That is the k-CAA problem is (t, ε)-difficult, which means there is no polynomial time algorithm to solve it with non-negligible probability ε.

Definition 6 (Random Masking): Our approach ensures the privacy of user data during the auditing process by employing a random masking v_i to hide μ, a linear combination of the data blocks.

B. CHALLENGE GAMES

The adaptive selection message attacking game adopted by this signature scheme is an interactive game between Challenger C and adversary $\mathcal{A}$. The game consists of the following three phases:

1) Setup. Challenger C obtains a new public and private key pair as (pk, sk) by the key generation algorithm. C sends the public key pk to $\mathcal{A}$ and saves sk.

2) Attack. The adversary $\mathcal{A}$ performs a polynomially bounded signature query. In the signature query, a message m is submitted to the challenger C. C runs the signature oracle O and returns the signature result $\sigma = Sign(sk, m)$ to $\mathcal{A}$.

3) Forgery. Adversary $\mathcal{A}$ forged the message m^* and its signature σ^*. The adversary will succeed if the following holds:

- The signature σ^* is a valid signature. Then, $Verify(pk, m^*, \sigma^*)$ does not return an error message.
- The adversary $\mathcal{A}$ did not ask a signature query for message m^*.

C. DATA INTEGRITY VERIFICATION SCHEME BASED ON BLS SIGNATURE

The BLS signature mechanism is a short message signature mechanism proposed by Boneh et al. [22]. Under the same security conditions, the signature bits of RSA, DSA, and BLS are 1024 bits, 320 bits, and 160 bits. The BLS signature mechanism requires a shorter number of signatures and can aggregate multiple signatures into one signature with good homomorphism. The PDP mechanism based on BLS signature supports public verification and meets the lightweight design requirements of cloud storage [32]. At present, there are many data integrity verification schemes based on BLS. Wang et al. [23] proposed a BLS-based data possession proof scheme. When the public verification was performed, the adversary could pass the verification with forged evidence without obtaining the user's data. In addition, the hash function $H(\mathbf{.}\ ): \{0,1\}^* \rightarrow G_1$ was used in that scheme. This function is a map-to-point (MTP, It was proposed by Boneh, Lynn, and Shacham in the literature [22]) hash function and needs to map the signed message to an element in the group G_1. The efficiency is low. Therefore, the secure hash function $H(\mathbf{.})$: $\{0,1\}^* \rightarrow Z_q^*$ is used in this paper. It can be a general cryptographic hash function such as MD5 or SHA-1. While ensuring security, signature efficiency is effectively improved. Please refer to section V for specific efficiency analysis.

IV. OUR PROPOSED DATA INTEGRITY VERIFICATION SCHEME

A. ZSS SIGNATURE

ZSS short signature is based on a bilinear pairing proposed by Zhang et al. [7]. The principle is consistent with BLS to construct a signature system that is difficult to CDH problem in group G, and the signature system is less overhead than BLS. We describe the ZSS signature as the following four phases:

1) ParamGen. Generate system parameters are $\{G_1, G_2, q, P, H, e\}$
2) KeyGen. Randomly selects $x \in z_q^*$ and computes $P_{pub} = xP$. P_{pub} is the public key and x is the private key.
3) Sign. Use private key x and a message m to computes signature $S = \frac{1}{H(m)+x}P$.
4) Ver. Given public key P_{pub}, message m, signature S, verify if $e(H(m)P + P_{pub}, S) = e(P, P)$. The verification works because of the following equations:

$$\begin{aligned} e(H(m)P + P_{pub}, S) &= e((H(m){+}x)P, (H(m){+}x)^{-1}P) \\ &= e(P, P)^{(H(m)+x).(H(m)+x)^{-1}} \\ &= e(P, P) \qquad (3) \end{aligned}$$

B. SYSTEM MODEL

The system model diagram in our scheme is shown in Figure 1. The model is mainly composed of clients, cloud storage servers and third-party auditors (TPA).

1) CLIENT

Mainly refers to users and data collection devices that have data storage requirements. The IoT's network control center stores the user data collected by the sensor on the cloud storage server provided by the CSP, and the client can establish communication with the cloud storage server.

2) CLOUD STORAGE SERVER

The computing resources, network resources, and storage resources provided by the cloud service provider CSP and be used to store user's data.

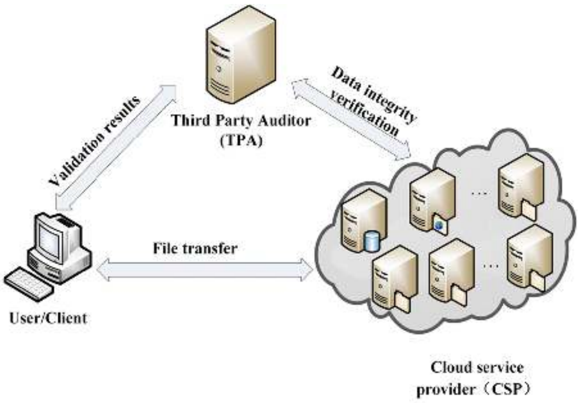

FIGURE 1. Data possession verification model.

3) THIRD-PARTY AUDITORS (TPA)

An independent, trusted third-party with expertise and capabilities. It doesn't know the stored data. After the user's authorization, instead of the user, it initiates data-possession verification to the cloud storage server and completes the data-possession verification or auditing.

In our model, users store data in the cloud storage server. In order to reduce storage costs, users will not save the original data locally. Therefore, the user needs to perform a data-possession verification in the cloud storage server to ensure data integrity and availability. In order to reduce the user's computational overhead and the communication overhead between the user and the cloud storage server, a third-party auditor TPA is introduced to instead of the user to implement the possessive verification, and only the verification result is returned to the user. Among that, TPA needs to complete the possession validation of data without obtaining user's data, to protect the data privacy. And the user is not affected by the complexity of the verification data, which make the verification easier and improve the efficiency. Cloud storage server stores multiple copies of data for multiple users. It is required to respond to the challenge initiated by the TPA and return evidence to the TPA.

C. BASIC DEFINITION OF THE SCHEME

Our scheme consists mainly of the following four algorithms:

KeyGen$(k) \rightarrow (pk,\ sk)$: Enter the security parameter k, output the user's public key pk and private key sk.

SigGen$(sk, m) \rightarrow \sigma$: Input a file m, a secret key sk, and output the signature set of the data block σ.

GenProof$(m, \sigma, chal) \rightarrow Pf$: It takes as inputs a file m, a signature collection σ of data blocks and the generated challenge message *chal*. It returns a proof of possession *Pf* for the specified data block that is determined by the challenge *chal*.

VerifyProof(*chal, Pf*)→{TRUE, FALSE}: It takes a challenge message *chal* and an integrity proof *Pf* as input, then, outputs the verification result TRUE or FALSE. If the verification passes, output TRUE, otherwise output FALSE.

D. THE BASIC SIGNATURE SCHEME

We define the system parameters in our scheme as follows. It is assumed that group G_1 is a cyclic additive group generated by P, G_2 is a cyclic multiplicative group. The order of G_1 and G_2 is q. Z_q denotes the integer ring of the mod q. Given bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$. Define a safe hash function H: $\{0, 1\}^* \rightarrow \{0, 1\}^{\lambda}$. Given $\phi(i, j)$: $Z_q^* \times \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$ is a pseudo-random function where $k_0 \in Z_q^*$ and $|q| \geq \lambda \geq 160$.

Our scheme has four stages: key generation, signature, challenge and verification.

1) KEY GENERATION

KeyGen()→(*pk, sk*). In the key generation stage, the client generates public key and private key. The user randomly selects $x \in Z_q^*$, calculates $Y = xP$, and gets the public key pk is Y and the private key sk is x.

2) SIGNATURE

SigGen(*sk, m*)→ σ. In the signing phase, the client generates the signature of data block. The file m is divided into data blocks $\{m_1, m_2, \ldots, m_n\}$, wherein the signature of the block m_i is [33]:

$$\sigma_i = \frac{1}{H(m_i) + x} P \tag{4}$$

Then, the signature of the file m is $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$. The client sends $\{m, \sigma\}$ to the CSP, sends σ to the TPA, and deletes the file m locally.

Challenges are generated by the TPA, and challenge messages are sent to the CSP. The TPA selects c elements from the set $\{1, 2, \ldots, n\}$ to form the set $I = \{s_1, s_2, \ldots, s_c\}$, where $1 \leq s_1 \leq \ldots \leq s_c \leq n$. For each $i \in I$, the TPA generates a pseudo-random number $v_i = \phi(k_0, i)$, and sends a challenge message $chal = \{(i, v_i)\}_{s_1 \leq i \leq s_c}$ to CSP.

3) CHALLENGE

GenProof$(m, \sigma, chal) \rightarrow Pf$. The CSP receives the challenge message *chal*, the signature $\{\sigma_i\}_{s_1 < i < s_c}$ of challenge data blocks, and then calculates:

$$R = \sum_{i=s_1}^{s_c} v_i Y \tag{5}$$

$$\mu = \sum_{i=s_1}^{s_c} v_i H(m_i) P \tag{6}$$

$$\eta = P - P^2 \sum_{i=s_1}^{s_c} \frac{v_i}{\sigma_i} \tag{7}$$

In Equations (5)-(7), Y is the public key, $Y = xP$, P is a public parameter and $P \in G_1$, v_i is the random number generated by TPA. After executing the evidence generation algorithm *GenProof*, CSP will send evidence $Pf = \{R, \mu, \eta\}$ to TPA.

4) VERIFICATION

VerifyProof(*chal, Pf*)→{TRUE, FALSE}. During this phase, after receiving the evidence *Pf*, the TPA verifies signature $\{\sigma_i\}_{s_1 < i < s_c}$ of data blocks. TPA verifies whether the challenge data block is correctly possessed by the following equations:

$$e(\eta, P) \cdot e(\mu + R, P) = e(P, P) \tag{8}$$

If the equation is true, outputs TRUE, otherwise, outputs FLASE.

E. GOAL ANALYSIS

1) PRIVACY PROTECTION

In our signature mechanism, H is an anti-collision hash function, so the adversary cannot get the data information through the signature $\sigma_i = \frac{1}{H(m_i)+x}P$. The adversary cannot obtain user's privacy information by intercepting the signature information, which effectively ensuring the privacy of the user data. In addition, compared with Zhang's ZSS [7] signature algorithm, we use the random masking technique during the signature and challenge phases to prevent the TPA from learning about the data; thus, preserving the user's (data) privacy.

2) PUBLIC AUDIT

On the premise of user authorization and privacy protection, by introducing a third-party auditor (TPA)to support public auditing in the data validation process. Instead of the user, TPA initiates data-possession verification to the cloud storage server and completes verification or auditing.

3) AUDITING CORRECTNESS

Our scheme can effectively prevent forgery attacks. That is, it can effectively prevent attackers who forge $Pf = \{R, \mu, \eta\}$ from passing the audit of TPA. Section V Theorem 3 provides specific proofs.

4) LIGHTWEIGHT

This solution shortens the signature time through the proposed short signature scheme, effectively reducing the client's computational overhead. It can effectively improve the efficiency of signatures in the scenario of multiple users, large data scale or larger number of data blocks. And by introducing a third-party auditor TPA instead of the user to implement the possessing verification, reduce the client's auditing overhead and the communication overhead in the verification process.

V. ANALYSIS OF THE SCHEME

A. CORRECTNESS ANALYSIS

Theorem 1: According to the provable data possession scheme proposed in this paper, if the TPA and CSP can reply and pass the data possession validation, the correctness of the scheme is proved.

Proof: According to the scheme of this paper, in the verification phase, if the message returned during the interaction between the TPA and the CSP is correct, then, the TPA receives the proof $Pf = \{R, \mu, \eta\}$ sent by CSP is correct. The calculation of the following TPA proves the correctness of this scheme.

$$\begin{aligned}
&e(\eta, P) \cdot e(\mu + R, P)\\
&= e\left(P - P^2 \sum_{i=s_1}^{s_c} \frac{v_i}{\sigma_i}, P\right) \cdot e\left(\sum_{i=s_1}^{s_c} v_i H(m_i)P + \sum_{i=s_1}^{s_c} v_i Y, P\right)\\
&= e\left(P - P^2 \sum_{i=s_1}^{s_c} \frac{v_i(H(m_i)+x)}{P}, P\right) \cdot\\
&\quad e\left(\sum_{i=s_1}^{s_c} v_i(H(m_i)+x)P, P\right)\\
&= e\left(-\sum_{i=s_1}^{s_c} v_i(H(m_i)+x)P, P\right) \cdot e(P, P) \cdot\\
&\quad e\left(\sum_{i=s_1}^{s_c} v_i(H(m_i)+x)P, P\right) = e(P, P) \qquad (9)
\end{aligned}$$

B. SECURITY ANALYSIS

Theorem 2: Given the evidence $Pf = \{R, \mu, \eta\}$ provided by CSP, TPA cannot recover user's data.

Proof: Assume that under the random oracle model, there is a simulator $\mathcal{P}$ that can generate a correct reply without obtaining user privacy data. Given TPA as an adversary $\mathcal{A}$. $\mathcal{A}$ simulates the input and output of the TPA by constructing a simulator $\mathcal{P}$. $\mathcal{P}$ possesses public information such as public key and tag values. $\mathcal{A}$ uses $\mathcal{P}$ by rewinding technology to crack the signature mechanism in this paper and obtain user's data information.

$\mathcal{A}$ inputs the set $\{s_1, s_2, \ldots, s_c\}$ and challenge information $\{R, \mu, \eta\}$, when the function *VerifyProof* outputs TRUE or FLASE, $\mathcal{A}$ outputs the result O'. Constructing the simulator $\mathcal{P}$ to do the following:

1)The simulator $\mathcal{P}$ chooses random numbers $i \in Z_q$ and $i \in \{1, 2, \ldots, n\}$.

2) $\mathcal{P}$ selects c elements from the set $\{1, 2, \ldots, n\}$ to form a new set $I = \{s_1, s_2, \ldots, s_c\}$.

3) $\mathcal{P}$ generates a challenge message $chal = \{(i, v_i)\}_{s_1 \le i \le s_c}$ and sends it to the CSP, then $\mathcal{P}$ outputs the result O'.

4) Assume that the correct data obtained by $\mathcal{A}$ is η, and the element R' is randomly selected in Z_q. Then input$In = \{I', chal, \eta, R'\}$, the $\mathcal{P}$ will output $Out = \{O', In\}$.

5)When verifying, $\mathcal{A}$ gets $\mathrm{In}_{\mathrm{T}} = \{I, chal, R', \eta, \mu'\}$ by inputting information.

When the CSP is trusted, $\mathcal{P}$ outputs $O' = 1$. In carrying out the possession validation, $\mathcal{P}$ uses $\eta = P - P^2 \sum_{i=s_1}^{s_c} \frac{v_i}{\sigma_i}, R'$ and μ' to challenge the data whether to be possessed correctly by verifying the equation $e(\eta, P) \cdot e\left(\mu' + R', P\right) = e(P, P)$. Where v_i is randomly assigned by the TPA. According to H is an anti-collision hash function in this signature mechanism, $\mathcal{A}$ cannot obtain data information through the signature $\sigma_i = (1/(H(m_i) + x))P$. Therefore, during the verification process, $\mathcal{P}$ cannot obtain user's privacy information. This effectively ensuring user's data privacy.

Theorem 3: If there is a $(t, q_H, q_S, \varepsilon)$-adversary $\mathcal{A}$ using adaptive selection message attack against our scheme, then there is a (t', ε')-algorithm $\mathcal{F}$ can solve q_S-*CAA*. That is, the probability of any adversary passing the integrity verification by forging signature evidence is negligible, when

$$\varepsilon' \geq \left(\frac{q_S}{q_H}\right)^{q_s} \cdot \left(\frac{q_S}{n}\right) \cdot \varepsilon \text{ and } t' = t.$$

Proof: For the signature mechanism proposed in this paper, the hash query $H\ (m_i)$ is performed before signing the data block m_i. Assume that $(t,\ q_H,\ q_S,\ \varepsilon)$- adversary $\mathcal{A}$ can used adaptive selection message attack method cracked the proposed signature scheme in this paper, and constructed a (t', ε')- algorithm $\mathcal{F}$ to solve q_S-CAA.

Suppose $\mathcal{F}$ does the following challenge: Given $P \in G_1$, $Q = xP, h_1, \ldots, h_{q_s} \in Z_q, (h_1+x)^{-1}P, \ldots,$ and $(h_{q_s}+x)^{-1}P$, to calculate $(h+x)^{-1}P$ for any $h \in \{h_1, \ldots., h_{q_s}\}$. $\mathcal{F}$ fakes the signer, uses public key pk = Q and replies hash function query and signature query. Then, challenges the following:

S$_1$: $\mathcal{F}$ prepares q_H replies $\{w_1, w_2, \ldots, w_{q_H}\}$ to the hash oracle and given $h_1, \ldots., h_{q_s}$ which are distributed randomly in the response set.

S$_2$: $\mathcal{A}$ performs hash oracle query on $m_i(1 \leq i \leq q_H)$. $\mathcal{F}$ sends the query result w_i to $\mathcal{A}$ and sends the challenge information *chal* to $\mathcal{A}$.

S$_3$: $\mathcal{A}$ makes a signature oracle query on w_i. If $w_i = h_j$, $\mathcal{F}$ returns a reply message $(h_j+x)^{-1}P$ to $\mathcal{A}$, otherwise the query is terminated. The probability of $\mathcal{F}$'s success is $P_1 \geq q_S/q_H$ at this stage.

S$_4$: $\mathcal{A}$ uses the response $(h_j+x)^{-1}P$ and the challenge *chal*, finally, $\mathcal{A}$ terminates the query and outputs the signature pair $\{m_i, \sigma_i\}_{1 \leq i \leq q_H}$. The hash value of m_i is w_l, and $\mathcal{A}$ calculates the evidence $Pf = \{R, \mu, \eta\}$ by the challenge. If it satisfies:

$$e(\eta, P) \cdot e(\mu + R, P) = e(P, P). \tag{10}$$

Then the signature pair $\{m_i, \sigma_i\}_{1 \leq i \leq q_H}$ is legally forged. At this time, $H(m_i) = w_l$ and $\sigma_i = (w_l + x)^{-1}P$, $\mathcal{A}$ outputs $\{w_l, \sigma_i\}$ as a prediction challenge for $\mathcal{F}$. The probability of $\mathcal{F}$ success at this stage is $P_2 \geq \frac{q_S}{q_H} \cdot \frac{q_H}{c} \cdot \frac{c}{n} = \frac{q_S}{n}$.

The signature oracle query of this scheme regards hash function as a random prediction. The running time of $\mathcal{F}$ is the same as $\mathcal{A}$, that is, $t' = t$. For all signature oracle queries, the probability of algorithm $\mathcal{F}$ succeeds after completing steps S$_3$ and S$_4$ is

$$\varepsilon' \geq (P_1)^{q_S} \cdot P_2 \cdot \varepsilon = \left(\frac{q_S}{q_H}\right)^{q_S} \cdot \left(\frac{q_S}{n}\right) \cdot \varepsilon.$$

C. PERFORMANCE ANALYSIS

This section mainly analyzes the computational overhead, storage overhead, and communication overhead. Experimental environment: With an Intel Core i7-4790 3.60 GHz workstation CPU, the memory size is 12G and the operating system is Windows7. In this paper, the elliptic curve domain is used to represent G_1 and G_2, the ECC key size is 160bit, and the random number size is 80bit. Experiments were carried out under the environment of PBC$-$0.4.7 and VC$++$6.0.

Suppose that file m is divided into n data blocks as $\{m_1, m_2, \ldots, m_n\}$, the provable data possession scheme based on the BLS and ZSS are compared between computational overhead and communication overhead.

1) COMPUTATIONAL OVERHEAD

In the BLS-based signature mechanism, we denote the addition on Z_q by Add_{Z_q}, the exponential operation in G_1 as Exp_{G_1}, the multiplication in Z_q as Mul_{Z_q}, and $Hash_{G_1}$ represents the Map-To-Point hash operation into the group G_1. The notation $MulExp_{G_1}$ represents multiplication operation n-term exponentiations like $\prod_{i=1}^{n} \sigma_i^{v_i}$, and $Pair_{G_1}, G_2$ denotes pairing operation $e(u_i, g_i)$, where $u_i \in G_1, g_i \in G_2$. The client computes the data blocks signature $\sigma_i = (H(m_i) \cdot u^{m_i})^x \in G_1$. Therefore, the client needs to perform n MTP operations as $nHash_{G_1}$. The server calculates a random factor $R = e(u, v)^r \in G_2$, an aggregated authenticator $\sigma = \prod_{i=1}^{n} \sigma_i^{v_i} \in G_1$, and a linear combination $\mu = \gamma \cdot \sum_{i=1}^{n} v_i m_i + r \in Z_q$,where $\gamma = h(R) \in Z_q$. Assume that the challenge message *chal* generated by TPA in the verification phase includes c random blocks. Under this setting, the computation cost of CSP is:

$$cMulExp_{G_1} + Exp_{G_2} + Hash_{Z_q} + cAdd_{Z_q} + (c+1)\,Mul_{Z_q}.$$

TPA needs to reply the evidence $Pf = \{R, \mu, \sigma\}$, and its computational cost is:

$$\begin{aligned} Hash_{Z_q} &+ cMulExp_{G_1} + cHash_{G_1} + 3Exp_{G_1} \\ &+ 2Pair_{G_1,G_2} + Mul_{G_1} + Mul_{G_2} \end{aligned}$$

Our scheme is based on ZSS signature mechanism and we use a general cryptographic hash function such as MD5, not a map-to-point (MTP) hash function like $Hash_{G_1}$ in BLS-based schemes. The client needs to perform n hash operations: n $Hash_{Z_q}$, the computation cost of the CSP is:

$$\begin{aligned} cHash_{Z_q} &+ 2cAdd_{Z_q} + (c+1)Add_{G_1} + cInvert_{Z_q} \\ &+ 4Mul_{G_1} + Mul_{Z_q}. \end{aligned}$$

The computation cost of the TPA is: $Mul_{G_2} + 2Pair_{G_1,G_1}$. Among them, the average computation time of $Hash_{G_1}$ is about 14.5ms, while the $Hash_{Z_q}$ is about 0.001ms. Obviously, $Hash_{G_1}$ requires more computing time than $Hash_{Z_q}$. The hash calculation results are compared as shown in Figure 2. The calculation time based on $Hash_{Z_q}$ is too small. In order to

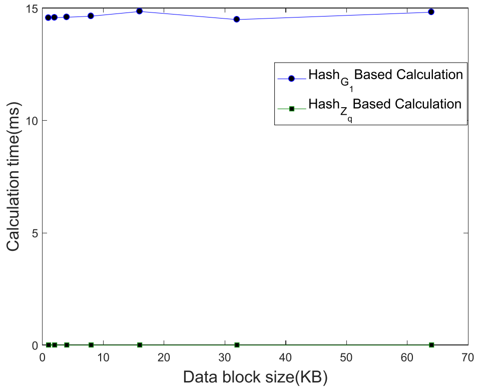

FIGURE 2. Operation time comparison for hash functions Hash$_{Z_q}$ and Hash$_{G_1}$.

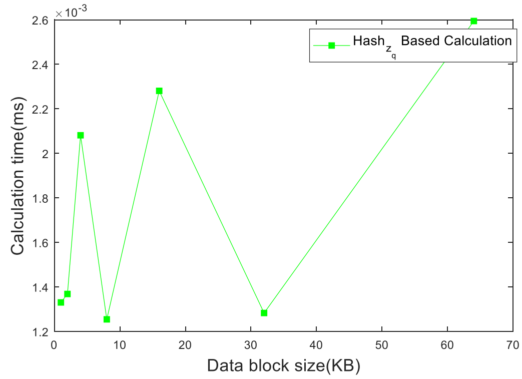

FIGURE 3. Operation time for hash functions $Hash_{Z_q}$.

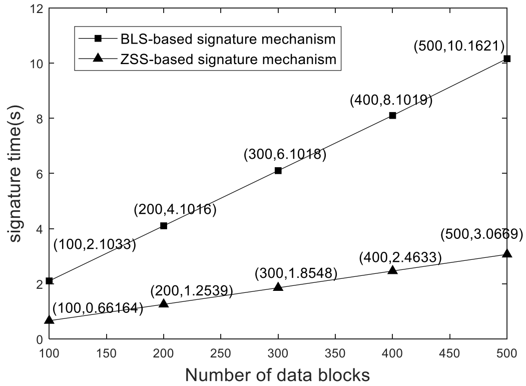

FIGURE 4. Signature time comparison for the two mechanisms.

make the reader see more clearly, the calculation time of $Hash_{Z_q}$ is shown in Figure 3.

$Hash_{G_1}$ denotes the Map-To-Point hash operation in the BLS mechanism, and $Hash_{Z_q}$ is the hash function used in this scheme. As shown in Figure 2 and Figure 3, the $Hash_{Z_q}$-based signature calculation cost is much lower than that of the $Hash_{G_1}$-based signature. In the signature process, $Hash_{Z_q}$ is used in this solution, and the computational overhead is significantly lower than that of the BLS schema.

When the user's data block size is fixed (for example, the size is 1 KB). As the number of data blocks increases, the signature time required for a signature scheme based on the ZSS takes less than that based on the BLS, as shown in Figure 4.

We use parameter $Mul\ G_1$ to denote the point scalar multiplication and use $Invert_{Z_q}$ denote the inversion in Z_q. The comparison of the computational overhead for these two schemes is shown in Table 1.

First, our scheme uses a generic hash function $Hash_{z_q}$: $\{0, 1\}^* \rightarrow Z_q^*$ in the signature phase, for example, MD5, while the hash function for BLS-based scheme is a MTP hash function $Hash_{G_1}$: $\{0, 1\}^* \rightarrow G_1$, which maps strings uniformly to the group G_1. The comparison of the computational overhead of these two hash functions is shown in Figure 2. Obviously, the computational overhead of using the MTP hash function $Hash_{G_1}$ is much larger than the hash function $Hash_{Z_q}$ we use. Therefore, as shown in Figure 4, in the signature phase, the computational cost of our scheme is smaller than that of BLS-based schemes. Second, our scheme uses Equation (8) to verify the integrity of the stored data. On the left side of Equation (8), there are two pairing, one multiplication and one addition operations are denoted as: $Mul_{G_2} + 2\ Pair_{G_1}, G_1 + Add_{Z_q}$, while on the right side of Equation (8) we have a pairing operation $e\,(P, P)$ can be pre-calculated after generating a public-private key pair, since P is a public parameter. Therefore, as shown in table 1, the main computational cost of our scheme in the verification phase is $2Pair_{G_1}, G_1 + cHash_{Z_q}$, and the number of pairing operations is the same as that in BLS-based scheme, but the overhead of hash calculation and power exponential calculation is significantly less than that of BLS-based scheme. So, through the above analysis, our solution is indeed more efficient than the BLS-based solution.

TABLE 1. Computational overhead comparison.

	BLS-based signature mechanism [23]	Our scheme
Client	$nHash_{G_1}$+$nMul_{G_1}$+$nExp_{Zq}$+ $nExp_{G_1}$	$nHash_{Zq}$+$nMul_{G_1}$+ $nInvert_{G_1}$
TPA	$cMulExp_{G_1}$+$cHash_{G_1}$+ $Hash_{Zq}$+3Exp_{G_1}+2$Pair_{G_1,G_2}$+ Mul_{G_1}+Mul_{Zq}	Mul_{G_2}+2$Pair_{G_1,G_1}$+Add_{Zq}
CSP	$cMulExp_{G_1}$+Exp_{G_2}+$Hash_{Zq}$+ $cAdd_{Zq}$+(c+1)Mul_{Zq}	$cHash_{Zq}$+2$cAdd_{Zq}$+Mul_{Zq} +4Mul_{G_1}+(c+1)Add_{G_1} +$cInvert_{Zq}$

2) COMMUNICATION OVERHEAD

The extra communication overhead of the scheme based on the BLS signature mechanism is about 960bits [23]. In the scheme based on the ZSS signature mechanism, the extra communication overhead generated by the client is mainly the signature value uploaded to the CSP, which is approximately 160 bits. The extra communication overhead for the TPA to initiate the challenge message *chal* to the CSP is approximately 80bits and the proof of possession $Pf = \{R, \mu, \eta\}$ sent by the CSP is approximately 480bits. Therefore, the extra communication overhead generated by the ZSS-based signature mechanism is approximately 720bits, which is less than the communication overhead required by the BLS-based signature mechanism.

3) STORAGE OVERHEAD

The scheme based on the BLS signature mechanism and the scheme based on the ZSS signature mechanism designed in this paper have the same storage overhead.

VI. CONCLUSIONS

The integration of cloud and IoT have become highly pervasive. Cloud-assisted Internet of Things will promote the development of "digitalization" and "intelligence" in human society. Aiming at the data security problem of cloud and IoT storage systems, this paper proposed new data integrity checking scheme combining ZSS signature.

The scheme fully considers security, scalability and privacy protection to meet the requirements of computing, communication and storage functions of data analysis applications with large amounts of aggregated data in the Internet of things. And our scheme has the following advantages:

1) A new remote data integrity verification scheme is implemented, which maintains the data integrity in IoT from the perspective of storage.

2) Using the ZSS signature algorithm, the computational overhead of the hash function is reduced than the BLS algorithm in the process of signature. Experiments show that this solution has less computational and communication overhead than existing RSA-based and BLS-based data integrity solutions.

3) Compared with Zhang's ZSS signature scheme [7], our scheme supports public auditing by introducing a trusted third party (TPA) and uses the random masking technique to preserving data privacy.

4) Based on the difficulty assumption of CDH, we prove that the scheme can resist adaptive selection message attack under the random oracle model.

However, compared with the BLS-based signature scheme in most existing cloud environments, this paper also has some shortcomings. For example, this scheme does not apply to data integrity verification in multiple replicas environments. Therefore, our next plan is to study a data integrity verification scheme that is more real-time and suitable for multi-copy environments. And to further enhance security by a signature mechanism that does not rely on the random oracle model.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 964–975, Jan. 2018.

[2] V. Adat and B. B. Gupta, "Security in Internet of Things: Issues, challenges, taxonomy, and architecture," *Telecommun. Syst.*, vol. 67, no. 3, pp. 423–441, 2018.

[3] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Bus. Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[4] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-based big data storage systems in cloud computing: Perspectives and challenges," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 75–87, Jan. 2017.

[5] C. Liu, C. Yang, X. Zhang, and J. Chen, "External integrity verification for outsourced big data in cloud and IoT: A big picture," *Future Gener. Comput. Syst.*, vol. 49, pp. 58–67, Aug. 2015.

[6] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for IoT data," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 468–475.

[7] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2004.

[8] M. Aazam, I. Khan, A. A. Alsaffar, and E. Huh, "Cloud of things: Integrating Internet of Things and cloud computing and the issues involved," in *Proc. 11th Int. Bhurban Conf. Appl. Sciences Technol. (IBCAST)*, Islamabad, Pakistan, Jan. 2014, pp. 414–419.

[9] G. Neagu, . Preda, A. Stanciu, and V. Florian, "A cloud-IoT based sensing service for health monitoring," in *Proc. E-Health Bioeng. Conf. (EHB)*, Jun. 2017, pp. 53–56.

[10] D. He, "Certificateless provable data possession scheme for cloud-based smart grid data management systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1232–1241, Mar. 2018.

[11] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," IACR Cryptol. ePrint Arch., HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2008-32, 2008, vol. 2008, p. 186.

[12] R. Chalse, A. Selokar, and A. Katara, "A new technique of data integrity for analysis of the cloud computing security," in *Proc. 5th Int. Conf. Comput. Intell. Commun. Netw.*, Sep. 2013, pp. 469–473.

[13] M. Venkatesh, M. R. Sumalatha, and C. SelvaKumar, "Improving public auditability, data possession in data storage security for cloud computing," in *Proc. Int. Conf. Recent Trends Inf. Technol.*, Apr. 2012, pp. 463–467.

[14] Y. Yu, L. Xue, M. H. Au, W. Susilo, J. Ni, Y. Zhang, A. V. Vasilakos, and J. Shen, "Cloud data integrity checking with an identity-based auditing mechanism from RSA," *Future Gener. Comput. Syst.*, vol. 62, pp. 85–91, Sep. 2016.

[15] Z. Xu, L. Wu, M. K. Khan, K.-R. Choo, and D. He, "A secure and efficient public auditing scheme using RSA algorithm for cloud storage," *J. Supercomput.*, vol. 73, no. 12, pp. 5285–5309, 2017.

[16] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "MuR-DPA: Top-down levelled multi-replica Merkle hash tree based secure public auditing for dynamic big data storage on cloud," *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2609–2622, Sep. 2015.

[17] A. Rajendran, V. Balasubramanian, and T. Mala, "Integrity verification using Identity based Provable Data Possession in multi storage cloud," in *Proc. Int. Conf. Comput. Intell. Data Sci. (ICCIDS)*, Jun. 2017, pp. 1–4.

[18] L. Chen, "Using algebraic signatures to check data possession in cloud storage," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1709–1715, 2013.

[19] S. Fu, D. Wang, M. Xu, and J. Ren, "Cryptanalysis of remote data integrity checking protocol proposed by L. Chen for cloud storage," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E97-A, no. 1, pp. 418–420, 2014.

[20] L. Yuchuan, F. Shaojing, X. Ming, and W. Dongsheng, "Enable data dynamics for algebraic signatures based remote data possession checking in the cloud storage," *China Commun.*, vol. 11, no. 11, pp. 114–124, Nov. 2014.

[21] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 847–859, May 2011.

[22] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2001, pp. 514–532.

[23] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.

[24] R. Mukundan, S. Madria, and M. Linderman, "Efficient integrity verification of replicated data in cloud using homomorphic encryption," *Distrib. Parallel Databases*, vol. 32, no. 4, pp. 507–534, 2014.

[25] J. Xu, L. Wei, W. Wu, A. Wang, Y. Zhang, and F. Zhou, "Privacy-preserving data integrity verification by using lightweight streaming authenticated data structures for healthcare cyber–physical system," *Future Gener. Comput. Syst.*, Apr. 2018. doi: 10.1016/j.future.2018.04.018.

[26] F. Rossi and G. Schmid, "Identity-based secure group communications using pairings," *Comput. Netw.*, vol. 89, pp. 32–43, Oct. 2015.

[27] Z. Qin, C. Yuan, Y. Wang, and H. Xiong, "On the security of two identity-based signature schemes based on pairings," *Inform. Process. Lett.*, vol. 116, no. 6, pp. 416–418, 2016.

[28] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1767–1777, Jun. 2018.

[29] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988.

[30] J.-L. Tsai and N.-W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Syst. J.*, vol. 9, no. 3, pp. 805–815, Sep. 2015.

[31] S. Mitsunari, R. Sakai, and M. Kasahara, "A new traitor tracing," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E85-A, no. 2, pp. 481–484, 2002.

[32] S. Tan, Y. Jia, and W. H. Han, "Research and development of provable data integrity in cloud storage," *Chin. J. Comput.*, vol. 38, no. 1, pp. 164–177, 2015.

[33] P. Kulshrestha and A. K. Pal, "A new secret handshakes scheme with dynamic matching based on ZSS," *Int. J. Netw. Secur. Appl.*, vol. 7, no. 1, pp. 67–78, 2015.