IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 14, 2020, accepted April 27, 2020, date of publication April 30, 2020, date of current version May 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991579

Lattice-Based Privacy-Preserving and Forward-Secure Cloud Storage Public Auditing Scheme

HAIFENG LI[1], LIANGLIANG LIU[2], CAIHUI LAN[3], CAIFEN WANG[4], AND HE GUO[1]

[1]School of Software, Dalian University of Technology, Dalian 116024, China
[2]School of Statistics and Information, Shanghai University of International Business and Economics, Shanghai 201620, China
[3]School of Electronic and Information Engineering, Lanzhou City University, Lanzhou 730070, China
[4]College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China

Corresponding author: Liangliang Liu (liangliang@suibe.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602080 and Grant 61602084, and in part by the Higher Education Institution Foundation of Gansu under Grant 2018A-112.

ABSTRACT Aiming at reducing the local storage burden and computational costs, numerous individuals and enterprises are willing to outsource their data to the cloud server. Meanwhile, due to the loss of the actual physical control over their data files once outsourced to the cloud server, how to guarantee the cloud server keep user's data integrity is an important security issue to be addressed urgently. Accordingly, multiple data integrity checking schemes based on the traditional cryptosystem have been proposed. However, with the advent and development of quantum computer, these existing data integrity checking schemes are no longer secure. Thus, it is necessary to study the new scheme which can resist quantum attack to adapt to the quantum era. In this work, we put forward a novel scheme named lattice-based privacy-preserving and forward-secure cloud storage public auditing scheme (LB-PPFS). Our proposed scheme is not only quantum-attack-against, but also enjoy the privacy-preserving and forward-secure property. In the proposed scheme, a curious auditor cannot learn any knowledge of user's data because the original data is encapsulated with a random number. In addition, the lattice basis delegation technique is adopted to achieve forward security for resisting key exposure attack. Based on the hardness assumptions of SIS problem from lattice, we prove that the proposed scheme can achieve formally provable security. Besides, the theoretical analysis and performance evaluation demonstrate that the proposed scheme is effective and feasible to guarantee the quantum security for the data integrity in cloud storage.

INDEX TERMS Cloud storage, public auditing, identity-based, lattice-assumptions.

I. INTRODUCTION

Nowadays numerous individuals and enterprises are willing to outsource their data to the cloud server in order to reduce the local storage burdens and computational costs. Meanwhile, considering the loss of physical control over data files, how to guarantee the cloud storage server to keep user's data intact becomes an urgent security issue to be addressed. Although being much stronger and more reliable than local devices, the cloud infrastructure is still subject to a wide range of threats from both internal and external adversaries on the integrity and availability of the outsourced data such as hardware device failure, system errors, software bugs and malicious hackers. For example, once an accident data corruption event occurs, the cloud storage provider may not inform the user this incident in time honestly only for the sake of maintaining its reputation, thus, making the user miss the golden opportunity to recover his valuable data. And what is even worse, the cloud storage provider may intentionally delete or alter the rarely accessed data of user so as to reclaim the storage space for maximizing its profit. Therefore, it is crucial for a client to perform efficient verification measures on the remote stored data periodically to ensure that their outsourced data not be modified or lost.

Data integrity verification in cloud storage has attracted intense interest because of its critical role in enhancing the

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

credibility of cloud service providers and the security of user data. Many researchers have investigated the security issue of cloud data integrity auditing and proposed various studies to resolve this problem [1]–[14].

A. RELATED WORK

In 2007, Ateniese *et al.* [1] first proposed the Provable Data Possession model for static data. The model implements data integrity checking by using RSA-Based Homomorphic Tags without downloading the entire data files from the cloud server to achieve blockless verification. In the PDP paradigm, the sampling method is used to perform the data integrity check, which improves the integrity verification efficiency significantly. With the rise and development of data outsourcing services, especially cloud storage services, more data integrity verification solutions begin to consider the actual application environment. Among these works, an attractive one is to support the public auditing feature. The public audit methodology enables the client to be free from heavy and dull burden on data integrity verification. To achieve public auditing, the client is required to hire a third-party auditor with professional expertise and strong computing power to periodically check the integrity of the data outsource to the cloud server on behalf of the data user. Wang *et al.* [14] put forward to a public auditing scheme with privacy-preserving property to against third party auditor by adopting the random masking technique.

The key exposure problem is another serious challenge to public auditing scheme. As is well known, the secret key is an indispensable and most crucial part of any cryptographic algorithm. In practice, the first core foundation task of deploying a cryptographic scheme is to secure storage and management of the secret key. Once the secret key is revealed, the whole cryptographic scheme will become completely insecure. However, the key-exposure problem is unavoidable in many real scenarios, including public auditing schemes for the outsourced data in the cloud storage server. First of all, in order to capture the secret key, the potential adversaries have exploited various attack methods, such as side-channel attacks [15], cold-boot attacks [16] and so on. Secondly, the sensitive secret key must be securely generated and stored in special well-protected cryptographic device, and delivered to the user by a secure channel. If the user carries the secret key by a non-special-trusted device (such as the common cheaper mobile phones or USB flash drive) or the transmission channel between the sender and receiver is not safe enough, the secret key is prone to be vulnerably exposed. Not to mention that the careless users and non-well-trained users may be lack of the sense of security and intentionally leak their private keys. The way by using the well-protected hard device to protect the secrete key from being exposed is costly and even impractical in some settings. Thus, researchers developed several ways without hardware to resist key exposure, such as secret sharing technique, forward security technique. The secret sharing technique require the user to split their secret(private) key into several components and distribute them to different participants resulting in high computation cost and communication cost. However, the forward security technique can provide the desired security in the presence of key exposure without distribution. Informal speaking, The idea of forward security is that the whole lifetime of a secret key is spitted into T distinct time periods enumerated by $1, 2, \cdots, T$. Accordingly, in each time period, user's private key can be denoted as $SK_1, SK_2, \cdots, SK_T$. The user will update(evolve) his/her secret key with time while the corresponding public key remains unchanged during the whole lifetime of a secret key. For instance, at the end of the i time period, the user updates the current SK_i to the new secret key SK_{i+1} which will be used in the next time period by using key update algorithm(the key update algorithm commonly is one-way function) with the current secret key SK_i as input, and then deletes the old secret key SK_i. In case of key exposure occurring in the current time period i, it means that if the secrete key SK_i is exposed to the adversary, but the prior time periods are not affected because the previous secret key $SK_1, SK_2, \cdots, SK_{i-1}$ have been deleted. In addition, the adversary cannot deduced the previous secrete key from the captured current key SK_i either because the SK_i is generated by a one-way function of the old secret key. Thus, the forward security technique can significantly mitigate the damage to the key exposure (since the leakage of the secret key in the time period i cannot compromise the security of the secret key in any previous time period). The formal definition of forward security is available in [17].

With respect to the public auditing scheme, the forward security technique is an effective solution to guarantee the security of the auditing key against key exposure and reduce the damage to the auditing key exposure to a minimal. However, most of the existing public auditing schemes are commonly designed in an idealized model and assume the secret key is safely kept. These schemes rarely consider the key-exposure problem in practice. Once the malicious cloud server captures the client's auditing secret key, it can succeed to pass the data verify auditing by forging the possession proof, therefore, the cloud can arbitrarily tamper with or discard the client's outsourced data for maximizing its economic profits. Apparently, the exposure of client's auditing secret key will cause fatal disaster to their outsourced data. Thus, how to protect client's secret key from being exposed or take effective measures to reduce the damage brought about by the client's secret key leaking to a minimum level deserve full consideration. Fortunately, in recent years, some cryptographic scholars begin to pay considerable attention to the key-exposure issue of the client's and many excellent works have been put forward to deal with it. As far as we know, Yu *et al.* studied the key-exposure issue in the cloud storage auditing for the first time [18]. Although their scheme [18] can significantly decrease the damage to key exposure, it also results in extra burden to the client because it is required the client to perform the key evolve algorithm in each time period. Subsequently, in order to reduce the client's computation cost, Yu *et al.* [8] proposed a new scheme which

supports verifiable outsourcing of key updates operation to an authorized party. However, scheme in [18] and [8] has a same security vulnerability that the adversary captures the secret key in time period i and not be detected in time, he/she will be able to continuously capture the evolved new secret key until the key exposure is found by the client. In order to end this problem, Yu and Wang [19] designed a strong key-exposure resilient auditing scheme. Based on [19], Ding *et al.* [6] strengthened the security of [19] by introducing the idea of intrusion-resilient. Besides these works, several attribute based encryption (ABE) schemes with leakage-resilience have recently been investigated in various application scenarios, such as hierarchical CP-ABE scheme with continuous leakage-resilience [20], KP-ABE scheme with continual auxiliary input leakage-resilience [21].

All the above-mentioned public auditing schemes are based on complicated Public Key Infrastructure (PKI). These PKI-based schemes suffer from the heavy and cumbersome certificates management and deployment of public key certificates to the client, especially troublesome to the resource-limited client such as mobile phone and iPad. In addition, it also brings about heavy burden of certificate verification for the third-party auditor. In order to eliminate the heavy burden of certificate management and verification in PKI model, Shamir [22] proposed a novel public key model called Identity-Based Cryptosystems in 1984. Wang *et al.* [23] proposed the first identity-based public auditing scheme in which the identity(e.g., telephone number, e-mail address, IP address) of client act as the public key and the corresponding secrete key of each client is extracted from the master private key of the Private Key Generator(PKG). Afterwards, they extended their identity-based public auditing scheme to multi-cloud setting [24]. Later, Peng *et al.* [25] found the security flaw of the scheme in [24] and presented a remedy solution. However, Lan *et al.* [26] pointed out their scheme also suffers a security vulnerability that the malicious cloud server can forge the data possess proof to pass the data integrity verification even without the client's original data. Accordingly, Lan *et al.* put forward a remedy solution to address this issue without changing the original security properties. Li *et al.* [9] proposed an identity-based PDP scheme for multi-cloud storage.

It is worth noting that most of the existing public auditing schemes will be broken completely by quantum computing when the quantum computer come into reality in the near future because the underlying difficult problems of them are the large integer factorization problem or the discrete logarithm problem, which will be solved by adopting the quantum computer in polynomial time according to the work of Shor [27]. Fortunately, lattice cryptography can provide us a promising solution to construct quantum-resistant cryptographic schemes. Following the Ajtai's creative work [28] in lattice, various lattice-based schemes have been proposed so far [29]–[36]. In the presence of the security challenge in quantum era, several cloud storage public auditing schemes from lattice [33]–[36] have been proposed. Public auditing schemes from lattice in [34], [35] are based on the complicated PKI model and these two schemes do not consider the data privacy security property, thus, the malicious auditor can obtain the information of the data block after multiple audits of the same data block. Although literature [33], [36] achieves the privacy preserving of user data, their schemes cannot resist forgery attack.

B. MOTIVATION AND CONTRIBUTION

The motivation of our work is described as follows.

Firstly, privacy-preserving and forward-security are two essential secure properties in public auditing schemes that aim to provide more secure and more reliable auditing for data integrity checking.

Considering that the third-party auditor is not fully trustworthy and he may be curious about the user's data information from the audit process with his powerful computation capacity, therefore, when designing a public auditing scheme it should be paid close attention to preserving privacy of user data, namely, public auditing scheme with privacy-preserving is required. In addition, considering that, in practice, due to various reasons discussed above(e.g., improper care, insecure hardware, Trojan virus), the user's secrete key may be leaked. Once an attacker has compromised user's secrete key, he can impersonate the authorized owner of the private key to do anything malicious things. Thus, when designing a public audit scheme, in order to protect the user's auditing secrete key from being compromised, public auditing scheme with key-exposure resilient by adopting forward security technique is also demanded.

Secondly, considering that most of exiting public auditing schemes cannot resist the potential quantum attack, therefore, when designing a public auditing scheme, it should be paid considerable attention to prepare for quantum security.

To summarize, it is necessary to design a lattice-based privacy-preserving and forward-secure cloud storage public auditing scheme (LB-PPFS) to provide quantum security in the quantum era.

The main contributions of this work can be summarized as follows:

1) The first contribution of this work is that we design an identity-based public auditing framework for cloud storage system and propose a lattice-based privacy-preserving and forward-secure cloud storage public auditing protocol. Our novel auditing protocol can guarantee the data privacy-preserving by using the random mask cryptography technique and achieve forward-secure property by using the lattice basis delegation technique.

2) The second contribution of this work lies in that we prove the security of our proposed LB-PPFS to demonstrate that it is provably secure under the hardness assumption of SIS and ISIS problem in random oracle model. Furthermore, we also conduct a performance analysis of the proposed scheme and compare it with that of previously proposed similar schemes to demonstrate that our proposed scheme is security resist quantum-attack and feasible in practice.

Thus, once the client initiates his request for data integrity checking to TPA, our protocol can achieve privacy-preserving verifying, forward-secure verifying, delegated verifying and public verifying of the integrity of the outsourced data.

C. PAPER ORGANIZATION

The rest of the paper is organized as follows. In Section II, we present preliminaries, including definitions and properties related to lattice, hardness assumption, basic system model and security definition. In Section III, we demonstrate the weakness of Zhang *et al.*'s scheme. In Section IV, we preset our concrete construction of LB-PPFS. In Section V, we give the correctness and security proven. In Section VI, we conduct the performance analysis of our proposal. Finally, we draw our conclusion in Section VII.

II. PRELIMINARIES

In this section, we give a brief review on the relevant knowledge of lattices, and introduce the basic system model and security definition of the proposed LB-PPFS.

A. LATTICES

Now we first review the definitions of lattices as follows.

Definition 1: Let $B = \{b_1, b_2, \cdots, b_m\} \in R^{m\times m}$ be a set of linearly independent column vectors, it generates an m-dimensional full-rank lattice Λ, which is defined as

$$\Lambda = \{Bc = \sum_{1\le i\le m} c_i b_i, c_i \in Z\}$$

Here $B = \{b_1, b_2, \cdots, b_m\}$ is a basis of the lattice Λ, the length of the basis B is the length of the longest vector in B, denote as $\|B\| = \max_{1\le i\le m} \|b_i\|$. Let $\tilde{B} = \{\tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_m\}$ be the Gram-Schmit orthogonalization of the vectors B taken in that order.

Definition 2: For q prime, matrix $A \in Z_q^{n\times m}$, the definition of the q-modular integer lattices are defined as follows.

1) $\Lambda_q(A) = \{x \in Z^m : \exists c \in Z_q^n, x = A^T c \bmod q\}$.
2) $\Lambda_q^{\perp}(A) = \{e \in Z^m : Ae = 0 \bmod q\}$.
3) $\Lambda_q^{y}(A) = \{e \in Z^m : Ae = y \bmod q\}$.

Lemma 1: For any prime $q \ge 2$, two positive integers n, m satisfies $m \ge \lceil 5n\log q\rceil$, there exists a probabilistic polynomial-time(PPT) algorithm *TrapGen*(q, n, m) [29] that returns a pair of $A \in Z_q^{n\times m}$ and $T \in Z_q^{m\times m}$, such that A is statistically close to a uniform matrix in $Z_q^{n\times m}$ and T is a basis for lattice $\Lambda_q^{\perp}(A)$ with $\|T\| \le O(n\log q)$ and $\left\|\tilde{T}\right\| \le O(\sqrt{n\log q})$.

Definition 3: The discrete Gaussian distribution over a subset L of Z^m with center $c \in R^m$ and Gaussian parameter $\delta > 0$ is $\forall x \in L$, $D_{L,\delta,c}(x) = \frac{\rho_{\delta,c}(x)}{\rho_{\delta,c}(L)}$. Where $\rho_{\delta,c}(x) = \exp(-\pi\|x\text{-}c\|^2/\delta^2)$ is called as Gaussian function, where $\rho_{\delta,c}(L) = \sum_{x\in L} \rho_{\delta,c}(x)$.

Lemma 2: Given any prime $q \ge 3$, positive integers $n, m \ge \lceil 2n\log q\rceil$ and a rank n matrix $A \in Z_q^{n\times m}$, there exists a PPT algorithm *SampleRwithBasis*(q, m, n, A) [32] that outputs an invertible and low-norm matrix sampled from a distribution statistically close to $R \in D_{m\times m}$, and a random short basis $T \in Z_q^{m\times m}$ of $\Lambda_q^{\perp}(AR^{-1})$, such that $\left\|\tilde{T}\right\| \le \sigma_R/\omega(\sqrt{\log m})$ with overwhelming probability.

Lemma 3: Given any prime $q \ge 2$, and two positive n, $m \ge \lceil 2n\log q\rceil$, there exists a PPT algorithm, denoted *Sample* $\Pr e(A, T_A, y, \delta)$ [29], that, on input a matrix $A \in Z_q^{n\times m}$, a short basis $T_A \in Z_q^{m\times m}$ of lattice $\Lambda_q^{\perp}(A)$, a vector $y \in Z_q^n$ and a Gaussian parameter $\delta \ge \left\|\tilde{T}_A\right\| \omega(\sqrt{\log m})$, it outputs a sample $\theta \in Z^m$ from a distribution within negligible statistical distance of $D_{\Lambda_q^y(A),\delta}$.

Now we recall the important lattice basis delegation technique proposed by Agrawal *et al.* in [32], which is used to realize the secrete key evolution algorithm to achieve the forward security property of our proposed scheme.

Lemma 4: There exists a PPT algorithm *NewBasisDel*(A, T_A, R, σ) [32], that, on input a matrix $A \in Z_q^{n\times m}$, a short basis $T_A \in Z_q^{m\times m}$ of lattice $\Lambda_q^{\perp}(A)$ and an invertible matrix $R \in Z_q^{m\times m}$ sampled from the distribution $D_{m\times m}$ which is defined as $(D_{Z^m,\sigma_R})^m$ and a Gaussian parameter $\sigma \ge \left\|\tilde{T}_A\right\| \sigma_R\sqrt{m}\omega(\sqrt{\log^3 m})$, outputs a randomized lattice basis T of $\Lambda_q^{\perp}(AR^{-1})$.

Definition 4 (Smoothing Parameter): For an m-dimensional lattice Λ and a real number $\varepsilon > 0$, its smoothing parameter $\eta_\varepsilon(\Lambda)$ is defined to be the smallest $s > 0$ meeting $\rho_{1/s}(\Lambda^*\backslash\{0\}) \le \varepsilon$.

Lemma 5: Λ_1, Λ_2 are two m-dimensional lattices, if $\Lambda_1 + \Lambda_2 = Z^m$ is satisfied, then for any a real number $\varepsilon \ge \eta_\varepsilon(\Lambda_1 \cap \Lambda_2)$ and a vector $c \in R^m$, the statistic distance between the distribution of $D_{\Lambda_1,\sigma,c} \bmod \Lambda_2$ and the uniform distribution on $(\Lambda_1 + \Lambda_2)/\Lambda_2$ is at most 2ε [37].

B. HARDNESS ASSUMPTION

Definition 5: The Small Integer Solution Problem(SIS) is described below. Given a prime q, a real number $\varsigma > 0$ and a matrix $A \in Z_q^{n\times m}$, to solve a nonzero integer vector $e \in Z^m$ such that $Ae = 0$ and $\|e\| \le \varsigma$.

Definition 6: The Inhomogeneous Small Integer Solution Problem(ISIS) is defined below. For a prime q, a real number $\varsigma > 0$, a matrix $A \in Z_q^{n\times m}$ and a vector $y \in Z_q^n$, to solve a nonzero integer vector $e \in Z^m$ such that $Ae = y$ and $\|e\| \le \varsigma$.

The main result of [29] is a connection between the hardness of the SIS, ISIS problems and the SIVP problem. For any poly-bounded $\varsigma = poly(n)$ and any prime $q > \varsigma\omega(\sqrt{n\log n})$, the average-case SIS, ISIS problems are as hard as approximating the SIVP problem in the worst case with certain factor $\varsigma\tilde{O}(\sqrt{n})$.

C. BASIC SYSTEM MODEL AND SECURITY DEFINITION

This section discusses the system model and security definition. As illustrated in Figure 1, the system framework of LB-PPFS in cloud computing is given, where the system framework consists of four different types of entities: a client,

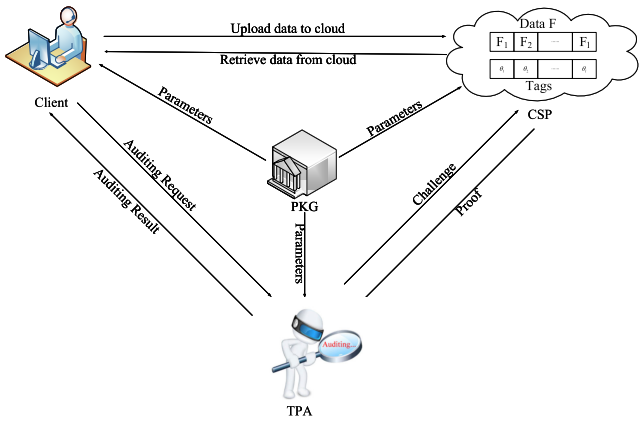


FIGURE 1. System model of the LB-PPFS.

CSP (the Cloud Service Provider), PKG (the Private Key Generator) and TPA (the Third-Party Auditor).

1) **Client.** The owner of the data, who has huge amount of data needed to move to the remote cloud server for storage, maintenance and sharing with others. It may be either individual consumer or organization.
2) **CSP.** It is an entity which has seemingly unlimited storage capability and computation ability and has the responsibility for storing and maintaining the outsourced data of the client. CSP is commonly regard as a semi-trusted party.
3) **PKG.** It is a fully trusted entity of the cloud storage system and takes the charge of generating the system parameters, public-private key and the private key for other entities, e.g., clients, CSP.
4) **TPA.** It is an independent third-party who has more expertise and capabilities than users and executes the data integrity verification on behalf of the data owner's request without learning the data content.

Definition 7 (Syntax): The syntax of the proposed LB-PPFS includes the following six-move algorithms as follows.

1) **Setup.** The algorithm performed by PKG and it outputs system parameters, master public key PK and master secret key SK according to the secure parameter and the total number of time periods r.
2) **Extract.** Given a client identity id_u and master public-secret key pair (PK, SK), this algorithm returns corresponding initial private key (first period key) $SK_{ID_u||1}$. Here $ID_u = id_u||r$.
3) **KeyUpdate.** Take as input current time period i, and the private key $SK_{ID_u||\tau}$ in a time period τ ($\tau < i$), this algorithm outputs the private key $SK_{ID_u||i}$ in the time period i.
4) **AuthGen.** Take as input current time period i, and the public-private key pair $(PK_{ID_u||i}, SK_{ID_u||i})$, and data file F, this algorithm outputs authentication ψ_i. At the same time, the client upload ψ_i and data file F to CSP, and removes them from the local storage completely.
5) **ProofGen.** The algorithm is performed by CSP and it outputs a response auditing proof $\mathrm{Pr}oof$ according to authentication ψ_i, data file F, and challenge information $chal$ be received from the auditor TPA.
6) **ProofVerify.** The TPA runs this algorithm to verify the validity of the proof $\mathrm{Pr}oof$ according to public PK and $PK_{ID_u||i}$, and challenge information $chal$. Finally, this algorithm outputs "True" if the $\mathrm{Pr}oof$ is correct that shows the data file is intact, otherwise, outputs "False" that indicates F is corrupted.

To continuously ensure the integrity of the data in the cloud, the auditor must periodically initiate the data integrity verification challenge to the cloud storage server. A secure public audit scheme should have no probabilistic polynomial time adversary that makes the auditor to accept the forged evidence with a non-negligible probability.

Following the security definition in [33], security model is described through a game played by an adversary and a challenger. To formalize the security model, the interaction of the game between an adversary and a challenger is defined as following.

Definition 8 (Security Model): Given security parameter, if there are no probabilistic polynomial adversaries who win the following games with a non-negligible probability, the LB-PPFS is select-ID and select-period secure. Here, let O be an adversary and C be a challenger.

1) **Setup:** The adversary O selects challenge identity ID_* and the time period i^*. The challenger C generates system public parameters and returns it to O.
2) **Queries phase:** The adversary O can adaptively query as follows.
   (a) **Private key query:** The adversary O adaptively queries the private key for any identity ID_u in any time period i. C calculates the relevant private key $SK_{ID_u||i}$ if $ID_u \neq ID_*$ or $i > i^*$, else sets the private key be $\perp$, and returns it to O.
   (b) **AuthGen query:** The adversary O adaptively selects data file $F = (F_1, F_2, \cdots, F_l)$ to query authentication for any identity ID_u in time period i. The challenger C computes corresponding authentication ψ_i of the data file F and sends it to O, then O stores ψ_i and F.
3) **Challenge:** The challenger C selects a specific challenge information $chal = \{L, v_i\}$ and sends it to O, where $L = \{l_1, \cdots, l_c\}$ is a subset of $\{1, \cdots, l\}$ and $v_i = \{v_{i,l_1}, \cdots, v_{i,l_c}\}$. O can continue to query for polynomial times as before, then outputs a response auditing forge proof $\mathrm{Pr}oof^*$ for the data files $F = (F_1, F_2, \cdots, F_l)$ indicated by $chal$. There are two following restrictions during the inquiry process.
   - The private key of ID_* cannot be queried in time period t $(1 \leq t \leq i^*)$
   - At least one of the data blocks F_{l_j} corresponding to $chal = \{L, v_i\}$ is not carried out **AuthGen query** for identity ID_* in time period i^*.

4) **Output:** The adversary O wins the above game if $\Pr oofVerify(PK_{ID_*\|i^*}, i^*, chal, \Pr oof^*) = 1$.

III. WEAKNESS OF ZHANG ET AL'S SCHEME

In this section, through the cryptoanalysis of Zhang *et al.*'s scheme [33], we demonstrate that their scheme has a security vulnerability that the malicious cloud servers could generate valid possession proof to pass the data integrity checking even without holding the original data of client. The specific analysis is described below.

Note that, based the concept of PDP, for one thing, in the ProofGen stage, among the data tuple (i, F, Ψ_i, ξ_i) which uploads to the cloud server for storing, only part component of them (i, Ψ_i, ξ_i) is used to generate data possession proof information and response to the challenge for data integrity checking and be verified by the TPA. For another thing, in the ProofVerify stage, the TPA performs to check the validity of the responded data auditing proof without requiring the data files themselves, therefore, in the ProofGen stage, the malicious cloud server may cheat TPA.

In the stage of AuthGen, the signature in Zhang *et al.*'s scheme [33] is related to A_c (A_c is the public key of the cloud server), that is, their scheme transforms signature of data file F_j to the signature of A_cF_j, i.e., the client adds an item A_cF_j. In the ProofGen stage, the cloud storage server picks a random vector w_i, and utilizes the SamplePre algorithm to generate a preimage β, in this way, the cloud storage server can encapsulate data file F_j with β, that is, $F_j' = \beta + F_j$. Then, In the ProofVerify stage, the TPA can recover A_cF_j by F_j'. However, this method is subjected to forgery risk that the malicious cloud storage server can run SamplePre algorithm to obtain the preimage of A_cF_j, that is, find another fake data F_j' such that $A_cF_j' = A_cF_j$. Meanwhile, since their scheme does not have a norm limit on F_j, the malicious cloud server can easily obtain the fake data F_j' by Gaussian elimination.

Although F_j' may be different from F_j by the above method, the malicious cloud server can generate valid proof information using F_j', namely, in the ProofGen stage, the malicious cloud server can substitute F_j with F_j' to generate the valid proof which can pass the data integrity checking in the ProofVerfy stage. Thus, Zhang *et al.*'s scheme [33] cannot resist forgery attack.

IV. THE CONCRETE LB-PPFS SCHEME

In this section, the concrete construction of our proposed LB-PPFS scheme for public auditing in cloud storage is presented in detail.

A. OVERVIEW

To facilitate the understanding of the proposed LB-PPFS scheme, in this subsection, we firstly present the sketch of our scheme. Its overall construction framework and workflow are demonstrated in Figure 2, which includes three stage.

1) Key generation stage. In this stage, the PKG is responsible for initializing the system parameters and extracting the corresponding secrete key for other entities in the system.

2) Authentication generation stage. The client generates the authentication for local data files, then uploads the data files and its corresponding authentications to the remote CSP.

3) Data audit stage. When receiving the data integrity audit request from client, the TPA generates the auditing challenge messages and sends them to the CSP. As a response, the CSP returns the corresponding data possess proof information for checking.

B. HIGH-LEVEL TECHNIQUE EXPLANATION

Our construction is based on two main key techniques: the lattice basis delegation technique and the random mask technique. In our design, we utilize the lattice basis delegation technique to achieve forward-secure property. As far as we are concerned, there exist several methods to implement the lattice basis delegation technique, such as literature [30], [31] and [32]. To the best of our knowledge, the lattice basis delegation technique NewBasisDel in [32] is one of the most promising techniques at present. Compared with [30] and [31], the lattice basis delegation technique in [32] has the characteristic of maintaining the fix lattice dimension upon delegation, which keeps the signature private key and signature length unchanged while the lattice basis delegation technique in [30] and [31] will expand the lattice dimensionality, thus doubling the size of the signature private key and the length of the signature. In practice, this will result in great increase of communication cost, which will reduce the efficiency of the system. To be specific, suppose that the current time period is i, the client can perform the NewBasisDel technique to updates(evolves) his/her current private $SK_{ID_u||i}$ to the new private key $SK_{ID_u||i+1}$ which will be used in the next time period in KeyUpdate algorithm with the current time period i and private key $SK_{ID_u||i}$ as input, and then deletes the old secret key $SK_{ID_u||i}$. In this way, the client updates(evolves) his/her private key with the whole time. Accordingly, taking as input the time period i and ID_u as input, the CSP, TPA can compute the corresponding public key $PK_{ID_u||i}$ themselves thereby removing the requirement of the complicated certificate management in PKI model. Therefore, firstly, the client utilize $SK_{ID_u||i}$ to generate the data authentication in the time period i, then uploading them to the CSP and the CSP can compute the corresponding public key $PK_{ID_u||i}$ to verify these data. When receiving the audit request from the client, the TPA can also compute the corresponding public key $PK_{ID_u||i}$ to perform the audit process to check whether the client's data is kept intact. In case of key exposure occurring in the current time period i, it means that if the private key $SK_{ID_u||i}$ is exposed to the adversary, but the prior time periods are not affected because the previous secret key $SK_{ID_u||1}, SK_{ID_u||2}, \cdots, SK_{ID_u||i-1}$ have been deleted. In addition, the adversary cannot deduced the previous secrete key from the captured current key $SK_{ID_u||i}$ either because the $SK_{ID_u||i}$ is generated by a one-way function of the old secret key.

In our design, we utilize the random mask technique to achieve the data privacy-preserving property. In order to

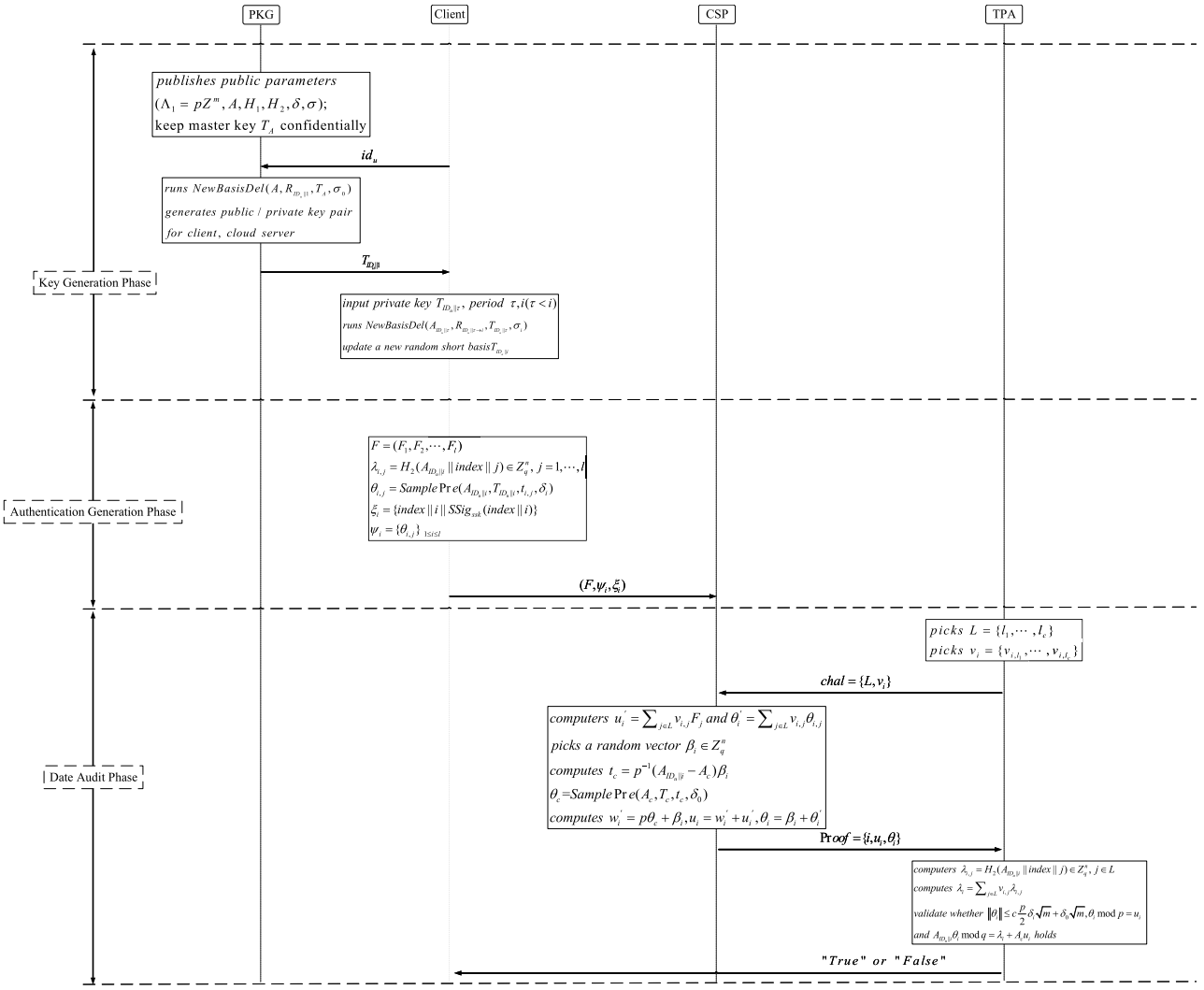


FIGURE 2. Overview of the LB-PPFS scheme.

prevent auditing proof from leaking any data information, that is, to provide a privacy-preserving auditing scheme, we modify the signature scheme in [38] to be related to both data F_j and A_cF_j (A_c is the public key of the cloud server), so the cloud storage server can randomly generate β_i such that $A_{ID_u||i}\beta_i = A_c w_i'$ when producing proof of possession for auditing, then encapsulate the signature with β_i and encapsulate the challenge data with w_i' by using the random mask technique. In this way, we can prevent the TPA from restoring the client's data after multiple queries.

C. CONSTRUCTION OF LB-PPFS

In this subsection, we present the detailed description of the proposed LB-PPFS scheme as follows.

1) **Setup:** Given the maximum number of challenge data blocks, the system establishment algorithm consists of the following six steps:

(a) The system selects two primes p, q such that $q \ge (mkp)^2$, and sets $n = \lfloor m/5 \log q \rfloor$.

(b) The system implements the algorithm *TrapGen* (q, m, n) to generate a matrix $A \in Z_q^{n \times m}$ and a short basis $T_q \in Z_q^{m \times m}$ of $\Lambda_q^{\perp}(A)$.

(c) The system calculates $T_A = pT_q$. Obviously, T_A is a good basis of lattice $\Lambda = pZ^m \cap \Lambda_q^{\perp}(A)$.

(d) The system defines two hash functions: $H_1 : \{0, 1\}^* \to Z_q^{m \times m}$ satisfies that its output value is a discrete Gaussian distribution (Standard deviation is σ_R) and $H_2 : Z_q^{n \times m} \times \{0, 1\}^* \to Z_q^n$.

(e) Given the number of period r, the system sets Gaussian parameters $\delta = (\delta_0, \delta_1, \delta_2, \cdots, \delta_r)$ and $\sigma = (\sigma_0, \sigma_1, \sigma_2, \cdots, \sigma_r)$ for the algorithms *Sample* Pre and *NewBasisDel* respectively. For any $0 \le i \le r$, $\delta_i \ge p\sqrt{n \log q}(\sigma_R\sqrt{m \log m})^i \log n$ and $\sigma_i \ge p\sqrt{n \log q}(\sigma_R\sqrt{m \log m})^{i+1} \log m$ are required. Obviously, it meets the conditions of Lemma 3 and Lemma 4.

(f) Finally, the system publishes public parameters $(\Lambda_1 = pZ^m, A, H_1, H_2, \delta, \sigma)$, and saves master key T_A confidentially.

2) **Extract:** On input a client identity id_u and public parameter, PKG runs $NewBasisDel(A, R_{ID_u||1}, T_A, \sigma_0)$ algorithm to obtain a random short basis $T_{ID_u||1} \in Z_q^{m\times m}$ of lattice $\Lambda_q^{\perp}(A_{ID_u||1})$ and sends it to the client through a secure channel. Here, $R_{ID_u||1} = H_1(ID_u||1)$ and $A_{ID_u||1} = A(R_{ID_u||1})^{-1} \in Z_q^{n\times m}$. In addition, the client needs to select a public/private pair (spk, ssk) for the signature of data file index. Similarly, PKG carries out the algorithm $NewBasisDel(A, R_{ID_c}, T_A, \sigma_0)$ to get a short basis T_c of lattice $\Lambda_q^{\perp}(A_c)$ for the cloud server. Here, $A_c = A(R_{ID_u||1})^{-1} = A(H_1(ID_c)^{-1})$.

3) **KeyUpdate:** Input public parameter, the client private key $T_{ID_u||\tau}$ in the previous period τ, and the current period i $(\tau < i < r)$, the client runs $NewBasisDel(A_{ID_u||\tau}, R_{ID_u||\tau\to i}, T_{ID_u||\tau}, \sigma_i)$ to get a random short basis $T_{ID_u||i} \in Z_q^{m\times m}$ of lattice $\Lambda_q^{\perp}(A_{ID_u||i})$. Here,

$$R_{ID_u||\tau\to i} = H_1(ID_u||i)\cdots H_1(ID_u||\tau+1)$$
$$R_{ID_u||\tau} = H_1(ID_u||\tau)\cdots H_1(ID_u||1),$$

and

$$A_{ID_u||\tau} = A(R_{ID_u||\tau})^{-1}$$
$$A_{ID_u||i} = A(R_{ID_u||i})^{-1} = A(R_{ID_u||\tau\to i}R_{ID_u||\tau})^{-1}.$$

4) **AuthGen:** The client divides the entire file F into l blocks and marked as $F = (F_1, F_2, \cdots, F_l)$, where $F_j \in Z_p^m$, then sets a unique index $index \in \{0,1\}^*$ of the data file. Finally, for each block F_j $(1 \le j \le l)$, the client computes its authentication $\theta_{i,j}$ according to the following steps, on input the current period i , the client public/private pair $(A_{ID_u||i}, T_{ID_u||i})$.

(a) Calculates l vectors $\lambda_{i,j} = H_2(A_{ID_u||i}||index||j) \in Z_q^n$, $j = 1, \cdots, l$.

(b) Solves $t_{i,j}$ such that

$$\begin{cases} t_{i,j} \bmod p = F_j \\ A_{ID*||i}t_{i,j} \bmod q = \lambda_{i,j} + A_cF_j \end{cases}$$

(c) Runs $Sample\,\Pr e(A_{ID_u||i}, T_{ID_u||i}, t_{i,j}, \delta_i)$ algorithm to obtain $\theta_{i,j}$.

(d) The client sends the data file F, the authentications $\psi_i = \{\theta_{i,j}\}_{1\le i\le l}$ and the signature $\xi_i = \{index||i||SSig_{ssk}(index||i)\}$ to cloud storage server, then deletes them from local storage. Here, ξ_i is used to ensure the integrity of the identity of data file.

5) **ProofGen:** Participants at this stage include the cloud server and the auditor TPA. The specific interaction process is as follows.

(a) When receiving the auditing request from the client, the TPA firstly obtains the data file tag ξ_i and verifies whether the integrity of the identity of data file is valid by using the client's public key spk to recover the signature $SSig_{ssk}(index||i)\}$. If the check is passed, then the TPA performs the following auditing steps. Otherwise, the TPA aborts the auditing process.

(b) The TPA generates challenge information $chal = \{L, v_i\}$, and send it to cloud storage server, where is the random subset $L = \{l_1, \cdots, l_c\}$ in the set $\{1, \cdots, l\}$ and $v_i = \{v_{i,l_1}, \cdots, v_{i,l_c}\} \in (-\frac{p}{2}, \frac{p}{2}]^{l_c}$.

(c) The cloud storage server receives $chal$, then computers $u_i' = \sum_{j\in L} v_{i,j}F_j$, $\theta_i' = \sum_{j\in L} v_{i,j}\theta_{i,j}$ and selects a random vector $\beta_i \in Z_q^n$ such that $\|\beta\|_i \le \delta_0\sqrt{m}$, and computes $t_c = p^{-1}(A_{ID_u||i} - A_c)\beta_i$.

(d) The cloud server runs $Sample\,\Pr e(A_c, T_c, t_c, \delta_0)$ algorithm to obtain θ_c and sets $w_i' = p\theta_c + \beta_i$.

(e) Finally, the cloud storage server transfers proof $\Pr oof = \{i, u_i, \theta_i\}$ to the auditor. Here, $u_i = w_i' + u_i'$, $\theta_i = \beta_i + \theta_i'$.

6) **ProofVerify:** TPA verifies the proof $\Pr oof = \{i, u_i, \theta_i\}$ by the following steps.

(a) Computes l vectors $\lambda_{i,j} = H_2(A_{ID_u||i}||index||j) \in Z_q^n$, $j \in L$.

(b) Computes $\lambda_i = \sum_{j\in L} v_{i,j}\lambda_{i,j}$.

(c) Verifies whether the following inequation $\|\theta_i\| \le c\frac{p}{2}\delta_i\sqrt{m} + \delta_0\sqrt{m}$ holds. If "True", continues to perform the following steps, Otherwise, stop.

(d) Checks whether the following equation $\theta_i \bmod p = u_i$ holds. If "True", continues. Otherwise, stop.

(e) Validates whether the following equation $A_{ID_u||i}\theta_i \bmod q = \lambda_i + A_cu_i$ holds. If it outputs "True" showed that the audit is passed. Otherwise, the audit is failed and return "False".

V. ANALYSIS OF THE PROPOSED SCHEME

A. CORRECTNESS PROOF

1) The inequations $0 < \|\theta_{i,j}\| \le \delta_i\sqrt{m}$ and $0 < \|\beta_i\| \le \delta_0\sqrt{m}$ hold according to Lemma 3, so $0 < \|\theta_i'\| \le c\frac{p}{2}\delta_i\sqrt{m}$ is satisfied. Thus, the following equation holds.

$$\|\theta_i\| = \|\theta_i' + \beta_i\| \le \|\theta_i'\| + \|\beta_i\| \le c\frac{p}{2}\delta_i\sqrt{m} + \delta_0\sqrt{m}$$

2) Sets $\Lambda_1 = pZ^m$, $\Lambda_{u,i} = \Lambda_q^{\perp}(A_{ID_u||i})$ and $\Lambda_i = \Lambda_1 \cap \Lambda_{u,i}$.

(a) Since $\theta_{i,j} \in \Lambda_i + t_{i,j}$, $\theta_i' = \sum_{j\in L} v_{i,j}\theta_{i,j} = \sum_{j\in L} v_{i,j}(\Lambda_i + t_{i,j}) = \Lambda_i + \sum_{j\in L} v_{i,j}t_{i,j}$ holds, further, $\theta_i' \bmod p = \sum_{j\in L} v_{i,j}t_{i,j} \bmod p = \sum_{j\in L} v_{i,j}F_j$ is correct.

(b) Also owing to $\Lambda_i \subset \Lambda_{u,i}$ holds, the following equation is correct.

$$A_{ID_u||i}\theta_i' = A_{ID_u||i}(\Lambda_i + \sum\nolimits_{j\in L} v_{i,j}t_{i,j}) \bmod q$$
$$= A_{ID_u||i}\sum\nolimits_{j\in L} v_{i,j}t_{i,j} \bmod q$$
$$= \sum\nolimits_{j\in L} v_{i,j}\lambda_{i,j} = \lambda_i$$

3) Since $\theta_i \bmod p = u_i$ is satisfied

(a) $\beta_i \bmod p = w_i'$

(b)

$$\begin{aligned} A_c w_i' \bmod q &= A_c(p\theta_c + \beta_i) \bmod q \\ &= (pt_c + A_c\beta_i) \bmod q \\ &= (pp^{-1}(A_{ID_u||i}\beta_i - A_c\beta_i) + A_c\beta_i) \\ &\quad \bmod q \\ &= A_{ID_u||i}\beta_i \bmod q. \end{aligned}$$

4) Thus, $\theta_i \bmod p = u_i$ and $A_{ID_u||i}\theta_i \bmod q = \lambda_i + A_c u_i$ are satisfied.

The scheme is correct according to above analysis.

B. SECURITY PROOF

In this subsection, we show that the proposed scheme is secure through the following two theorems.

Theorem 1: The proposed LB-PPFS scheme achieves forward security, provided that the hardness assumption of SIS problem is intractable.

Proof: Given a challenge matrix $B \in Z_q^{n\times m}$, the challenger C obtains a vector $\theta_{i,t}^* \in Z_q^m$ such that $B\theta_{i,t}^* = 0$ and $0 < \left\|\theta_{i,t}^*\right\| \le \delta_* \sqrt{m}$ by playing game (Definition 8) with an adversary O. At the beginning of the game, O determines the challenge identity id_* and the challenge time period i^*, and C maintains two empty lists L_1, L_2. C adopts the algorithm *SampleRwithBasis* to generate $R_*^1, R_*^2, \cdots, R_*^{i^*} \leftarrow D_{m\times m}$, then calculates $A = BR_*^{i^*}R_*^{i^*-1}\cdots R_*^1$, $A_{ID_*||1} = A_1(R_*^1)^{-1}$ and $A_{ID_*||j} = A_{ID_*||j-1}(R_*^j)^{-1}$ for $1 < j \le i^*$, and records $(ID_*, 1, A_{ID_*||1}, R_*^1, \perp)$, $\cdots$, $(ID_*, 1, A_{ID_*||i^*}, R_*^{i^*}, \perp)$ into list L_1. Finally, the challenger returns the system parameter $(A, H_1, H_2, spk, \delta, \sigma)$.

Hash Queries: The adversary O can perform H_1, H_2 inquiries at any time as follows.

$H_1(id_u||i)$ Query: Given an identity id_u and a period i, it returns R_i if the pair (id_u, i) is found in list L_1. Otherwise, it calculates and returns the relevant hash value according to the following five conditions.

(a) When $id_u = id_*$ and $i = i^* + 1$, the challenger C performs the algorithm *SampleRwithBasis*$(A_{ID_*||i})$ to get a matrix $R_{i^*+1} \leftarrow D_{m\times m}$ and a short basis $T_{ID_*||i+1}$ of the lattice $\Lambda_q^{\perp}(A_{ID_*||i+1})$, where $A_{ID_*||i+1} = A_{ID_*||i}(R_{i^*+1})^{-1}$. The challenger C appends $(id_u, i^*+1, A_{ID_*||i^*+1}, R_{i^*+1}, T_{ID_*||i^*+1})$ into list L_1 and return R_{i^*+1}.

(b) When $id_u = id_*$ and $i > i^* + 1$, the challenger C first obtains the private key $T_{ID_*||i-1}$ of identity id_u in the time period $i-1$ by running $H_1(id_u||i-1)$ query, then selects a matrix $R_i \leftarrow D_{m\times m}$ and generates a short basis $T_{ID_u||i}$ of the lattice $\Lambda_q^{\perp}(A_{ID_u||i})$ by carrying out the algorithm *NewBasisDel*$(A_{ID_u||i-1}, R_i, T_{ID_u||i-1}, \sigma_i)$. Finally, C appends $(id_u, i, A_{ID_u||i}, R_i, T_{ID_u||i})$ into list L_1 and returns R_i. Here, $A_{ID_u||i} = A_{ID_u||i-1}(R_i)^{-1}$.

(c) When $id_u \neq id_*$ and $i = 1$, the challenger C runs the algorithm *SampleRwithBasis*(A) to get a matrix $R_i \leftarrow D_{m\times m}$ and a short basis $T_{ID_u||1}$ of the lattice $\Lambda_q^{\perp}(A_{ID_u||1})$, where $A_{ID_u||1} = A(R_i)^{-1}$. The challenger C appends $(id_u, 1, A_{ID_u||1}, R_i, T_{ID_u||1})$ into list L_1 and outputs R_i.

(d) When $id_u \neq id_*$ and $i > 1$, the challenger C executes $H_1(id_u||i-1)$ query to get the private key $T_{ID_u||i-1}$ of identity id_u in time period $i-1$, then selects a matrix $R_i \leftarrow D_{m\times m}$ and computes $A_{ID_u||i} = A_{ID_u||i-1}(R_i)^{-1}$, further performs the algorithm *NewBasisDel*$(A_{ID_u||i-1}, R_i, T_{ID_u||i-1}, \sigma_i)$ to generate a short basis $T_{ID_u||i}$ of lattice $\Lambda_q^{\perp}(A_{ID_u||i})$. Finally, C appends $(ID_u, i, A_{ID_u||i}, R_i, T_{ID_u||i})$ into list L_1 and returns R_i.

$H_2(A_{ID_u||i}||index||j)$ Query: Given $A_{ID_u||i}, index, j$, C returns $\lambda_{i,j}$ if pair $(A_{ID_u||i}, index, j)$ is found in list L_2. Otherwise, C randomly selects $f \in Z_p^m$ and samples a spot $\lambda_{i,j}'$ in D_{pZ^m+f,δ_i}, and sets $\lambda_{i,j} = H_2(A_{ID_u||i}||index||j) = A_{ID_u||i}\lambda_{i,j}' \bmod q - A_c f$ and returns it. Finally, the challenger C records $(A_{ID_u||i}, index, j, \lambda_{i,j}', \lambda_{i,j})$ into list L_2.

Private Key Query: Given an identity id_u and a time period i, the challenger C returns $SK_{ID_u||i} = T_{ID_u||i}$ by performing the query of $H_1(id_u||i)$.

AuthGen Query: Given a data file $F = (F_1, F_2, \cdots, F_l)$, an identity id_u and a period i, the challenger C calculates and returns its authentication of the data file F according to the following two conditions.

(a) When $id_u = id_*$ and $i \le i^*$. The simulator is failure if $H_2(A_{ID_u||i}||index||j)$ was queried before. Otherwise, For $1 \le j \le l$, the challenger C samples a spot $\lambda_{i,j}'$ in $D_{pZ^m+F_j,\delta_i}$, and sets $\lambda_{i,j} = H_2(A_{ID_u||i}||index||j) = \lambda_{i,j}' \bmod \Lambda_q^{\perp}(A_{ID_u||i}) - A_c F_j$. The challenger C appends $(A_{ID_u||i}, index, j, \lambda_{i,j}', \lambda_{i,j})$ into list L_2 and returns $\lambda_{i,j}'$ and $\xi_i = \{index||i||SSig_{ssk}(index||i)\}$.

(b) Others, the challenger C initiates private key query to obtain $SK_{ID_u||i} = T_{ID_u||i}$, and then performs the AuthGen algorithm to get authentication.

Challenge: After the above inquiry, the challenger C selects a specific challenge message *chal* of data file $F^* = (F_1{}^*, F_2{}^*, \cdots, F_l{}^*)$ and sends it to O. Finally, the adversary O outputs a forage proof $\Pr oof^* = \{i^*, u_{i^*}, \theta_{i^*}\}$.

Output: After obtaining the forged proof $\Pr oof^* = \{i^*, u_{i^*}, \theta_{i^*}\}$, the challenger C extracts u_{i^*} and θ_{i^*}. C obtain authentication $\psi_i = \{\theta_{i,1}, \theta_{i,2}, \cdots, \theta_{i,l}\}$ by looking up the lists L_1, L_2 and running AuthGen query. Further, C gets another proof $\Pr oof^* = \{i^*, u_i, \theta_i\}$ by performing the algorithm ProofGen, where $\theta_i = \sum_{j\in chal} v_{i,j}\theta_{i,j} + \beta_i$, $u_i = \sum_{j\in chal} v_{i,j}F_j + w_i'$, β_i is a random number. Obviously, there are $\theta_i \neq \theta_{i^*}$ and $A_{ID_*||i^*} = BR_*^{i^*}R_*^{i^*-1}\cdots R_*^1(R_*^1)^{-1}\cdots(R_*^{i^*-1})^{-1}(R_*^{i^*})^{-1} = B$. The equation $B(\theta_i - \theta_{i^*}) \bmod q = 0$ holds because of the following equation

$$\begin{aligned} A_{ID_*||i^*}\theta_i &= \sum\nolimits_{j\in chal} v_{i,j}H_2(A_{ID_*||i^*}||index||j) \\ &\quad + A_{ID_*||i^*}H_2(A_C||w_{i^*}||1) \\ &= A_{ID_*||i^*}\theta_{i^*} \bmod q\,. \end{aligned}$$

Furthermore, since $\left\|\theta_i\right\|$ and $\left\|\theta_{i^*}\right\|$ are no more than $c\frac{p}{2}\delta_i\sqrt{m} + \delta_0\sqrt{m}$, $\left\|\theta_i - \theta_{i^*}\right\| \le cp\delta_i\sqrt{m} + 2\delta_0\sqrt{m}$ holds. As mentioned above, literature [29] is a bridge between the

hardness of the SIS(ISIS) problems and the SIVP problem, therefore, the parameters $q \geq (mkp)^2$, $n = \lfloor m/5 \log q \rfloor$ and $\delta_i \geq p\sqrt{n \log q}(\sigma_R \sqrt{m} \log m)^i \log n$ are set in our proposed scheme, that meets the parameters requirement which make SIS problem be difficult.

During the simulation, since files tag *index* is selected randomly, the abortion probability is negligible at the stage of AuthGen query. Here, define two distributions of authentication. One is generated in the above game, that selects spot $\theta_{i,j} = \lambda_{i,j}{}'$ in $D_{pZ^m+F_j,\delta_i}$, and sets $H_2(A_{ID_u||i}||index||j) = \lambda_{i,j}{}' \mod \Lambda_q^{\perp}(A_{ID_u||i}) - A_c F_j$. The other is generated in proposed scheme, that solves $t_{i,j}$ according to the equation set

$$\begin{cases} t_{i,j} \bmod p = F_j \\ A_{ID_*||i} t_{i,j} \bmod q = \lambda_{i,j} + A_c F_j \end{cases}$$

then computes

$$\theta_{i,j} = Sample\Pr e(A_{ID_u||i}, T_{ID_u||i}, t_{i,j}, \delta_i).$$

We know the above two distributions are indistinguishable. According to lemma 5, the proposed LB-PPFS scheme achieves forward security.

Theorem 2: The proposed LB-PPFS scheme achieves data privacy preserving against the curious TPA, provided that the hardness assumption of SIS problem is intractable.

Proof: To save space, here we omit detailed description of the game in Theorem 2, but there are some key points should be pointed out as follows.

Taking into count that $u_i{}' = \sum_{j \in L} v_{i,j} F_j$ is a linear combination of data blocks which is sampled in the challenge information *chal*, the curious TPA may attempt to recover the original data blocks of the client by taking advantage of its powerful computation. To handle the security vulnerability of privacy leakage, in the stage of ProofGen, CSP generates a vector β_i that satisfies $\|\beta_i\| \leq \delta_0 \sqrt{m}$ by random sampling technique, and encapsulates θ_i into $\theta_i{}'$ with β_i, which could prevent TPA to recover $\theta_i{}'$ from θ_i, and further obtain $u_i{}'$ by using the equation $\theta_i{}' \mod p = u_i{}'$. Meanwhile, CSP utilizes $SamplePre(A_c, T_c, t_c, \delta_0)$ to compute θ_c and encapsulates $u_i{}'$ into u_i with θ_c, which could prevent TPA to recover $u_i{}'$ from u_i. In order to successfully solve these linear combinations, the curious TPA must obtain the valid θ_c. Therefore, if the adversary has a nonnegligible probability to compute θ_c it means that the adversary can succeed to solving the hardness assumption of ISIS problem $T_c \theta_c = t_c$. This is a contradiction because based on the security proof in [29], without knowing the trapdoor T_c of the CSP, the TPA can only solve the θ_c with negligible probability. Thus, we can safely draw the conclusion that TPA can't learn the knowledge of user data from the auditing process. Therefore, our proposed scheme preserves privacy against the curious TPA.

VI. PERFORMANCE ANALYSIS

In this section, we first present the functionality comparison of among the proposed LB-PPFS scheme and other several existing relevant schemes [8], [14], [33]. After that, we compare computation cost between our Scheme and Zhang *et al.*'s scheme [33] in terms of AuthGen stage, ProofGen stage and ProofVerify stage, respectively. At last, we discuss the performance comparison between Zhang *et al.*'s scheme [33] and our scheme in experiments.

A. FUNCTIONALITY COMPARISON

In this subsection, we present a summary on the functionality comparison between our LB-PPFS scheme and several existing schemes [8], [14], [33] with respect to the functionalities of storage correctness, blockless verification, probabilistic sampling, public auditability, data privacy preserving, forward security, post-quantum security in Table 1.

From Table 1, it is obviously observed that all the schemes can support blockless verification, probabilistic sampling, public auditability. Wang *et al.*'s scheme [14] can achieve data privacy preserving while not supporting the security property of forward security, post-quantum security.Yu *et al.*'s scheme [8] can achieve data privacy preserving and forward security but can not provide post-quantum security either. Zhang *et al.*'s scheme [33] can support the security property of data privacy preserving, forward security and post-quantum security, but fail to achieve storage correctness which has been discussed above. Only our LB-PPFS scheme can support all the following security properties: storage correctness, blockless verification, probabilistic sampling, public auditability, data privacy preserving, forward security, post-quantum security.

Based on the above comprehensive comparison, we could draw a concrete conclusion that our LB-PPFS scheme can achieve data privacy preserving and forward security public auditing as well as provide the post-quantum security simultaneously.

B. PERFORMANCE COMPARISON

According to the efficiency of Boneh and Freeman's signature scheme in [38], the auditing scheme we constructed is also effective since it is designed based on their signature scheme. The following is a specific comparison with the identity-based auditing scheme proposed by Zhang *et al.* [33]. For the sake of comparison, we summarize the relevant parameters n, m, q, σ, δ in our scheme and their meaning in Table 2.

1) The main calculations include $n+l$ hash operations, $2nl$ inner products(i.e., nl inner product of m-dimension vector and nl inner product of n-dimension vector), l preimage samples and one signature in Zhang *et al.*'s scheme. In our scheme, the primary computation including l hash operations, l preimage samples, one signature and solving the following equations l times.

$$\begin{cases} t_{i,j} \bmod p = F_j \\ A_{ID_u||i} t_{i,j} \bmod q = \lambda_{i,j} + A_c F_j \end{cases} \quad (1)$$

Obviously, Let $t_{i,j} = F_j + pX$, $\tilde{\lambda}_{i,j} = \lambda_{i,j} + A_c F_j$, then the formula (1) can be transformed to $A'X \mod q = p^{-1}B^{-1}\tilde{\lambda}_{i,j} - p^{-1}A'F_j$, where A' is the simplest row corre-

TABLE 1. Functionality comparison.

Schemes	Storage correctness	Blockless verification	Probabilistic sampling	Public auditability	Data privacy preserving	Forward security	Post-quantum security
Wang et al.'s scheme [14]	√	√	√	√	√	×	×
Yu et al.'s scheme [8]	√	√	√	√	√	×	×
Zhang et al.'s scheme [33]	×	√	√	√	√	√	√
Our proposed scheme	√	√	√	√	√	√	√

√: deontes that the specified scheme is secure;
×: denotes that the specified scheme is insecure.

TABLE 2. Parameters and their meaning.

Parameter	Meaning
n	security parameter
m	satisfy $m \geq \lceil 5n \log q \rceil$
q	a prime number with $q \geq 2$
δ	Gaussian parameter for SamplePre algorithm
σ	Gaussian parameter for NewBasisDel algorithm

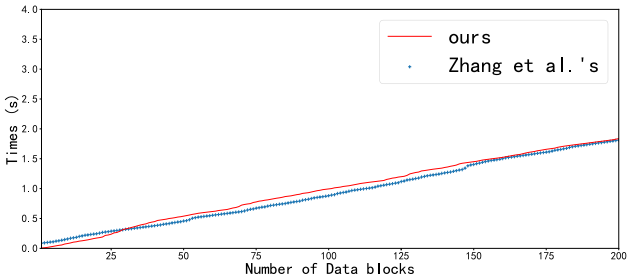


FIGURE 3. Performance Comparison in AuthGen.

sponding to matrix $A_{ID_u||i}$, B is a reversible matrix and such that $BA_{ID_u||i} = A'$, and $X \in Z^m$ meets $t_{ij} = F_j + pX$. The client ID_u can precompute A', $p^{-1}B^{-1}$, $p^{-1}A'$ and store, so the time of solving the equations approximates $2n$ inner products operations, i.e., nl inner product of n-dimension vectors.

2) In the stage of **ProofGen**. Based on the above time analysis of solving equations, our scheme has n inner products operations than Zhang *et al.*'s.

3) In the stage of **ProofVerify**. The main calculations include $n + c + l$ hash operations and $3n$ inner products in Zhang *et al.*'s scheme. And our scheme needs c hash operations and $2n$ inner products of m-dimension vectors.

Table 3 gives a summary of the comparison of computation cost between our scheme and Zhang *et al.*'s scheme [33] of three stage in terms of AuthGen stage,ProofGen stage and ProofVerify stage, where T_{ha}, T_{mu}, T_{sam}, T_{sig} denotes hash operation, multiplication operation, SamplePre operation and SSig signature operation, respectively.

Now, we discuss the comparison of the performance time of three different stage between the Zhang *et al.*'s scheme [33] and our proposed scheme which is shown as Figure 3, Figure 4 and Figure 5. Our experiment simulation is conducted with Python 3.7 and the system platform is windows 7 ultimate with the Intel(R)Core (TM) 4130 CPU @3.40GHz,4 GB RAM. In order to achieve 80 bit-security, we set the

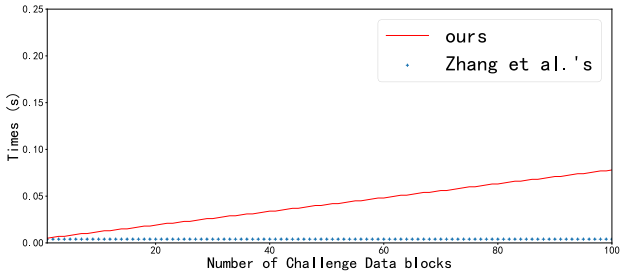


FIGURE 4. Performance Comparison in ProofGen.

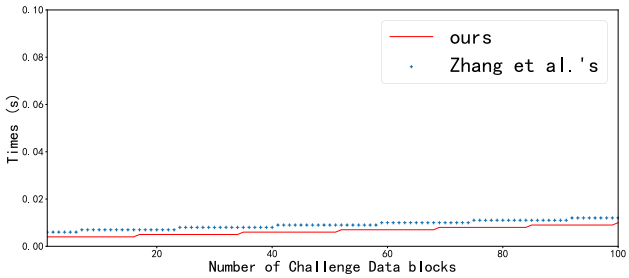


FIGURE 5. Performance Comparison in ProofVerify.

parameters as $n = 64, m = 11978, q = 2^{54} - 33, p = 127, r = 10, k = 100$. Here each experiment is performed 30 times, then we show the average computation cost in these figures.

Figure 3 shows that the computation cost at the stage of AuthGen for Clients grows linearly with the amount of the data blocks I. The computation costs in our scheme are similar to Zhang *et al.*'s scheme [33] at the AuthGen stage.

Figure 4 shows that the computation cost at the stage of ProofGen for CSP grows linearly with the number of challenged data blocks from TPA. It can be seen that the computational cost in Zhang *et al.*'s scheme [33] is nearly to a constant, which almost has nothing to do with the number of challenged data blocks, but the computational cost in our scheme keeps in a low level.

Figure 5 shows that the computation cost at the stage of ProofVerify for TPA grows linearly with the number of the challenge data block. It can be observed that both schemes are done well, but our scheme is slightly outperformed than Zhang *et al.*'s.

From the above performance analysis, it can be reached that the overall computation cost of our proposed scheme

TABLE 3. Comparison of computation cost between our scheme and Zhang *et al.*'s Scheme [33].

Scheme	AuthGen	ProofGen	ProofVerify
Zhang et al.'s [33]	$(n+l)T_{ha}+(nml+n^2l)T_{mu}+T_{sam}+T_{sig}$	$T_{ha}+mT_{mu}+T_{sam}$	$(n+c+1)T_{ha}+(cn+2mn+n^2+2n)T_{mu}$
Our scheme	$lT_{ha}+(2nml+n^2l)T_{mu}+T_{sam}+T_{sig}$	$(nm+2n+mc)T_{mu}+T_{sam}$	$cT_{ha}+(2nm+nc)T_{mu}$

is almost similar to that of [33], while our scheme provides privacy-preserving and forward-secure guarantee, and overcomes the security vulnerability of [33] that CSP could generate valid proof even without the original data files.

VII. CONCLUSION

In this work, we propose a novel privacy-preserving and forward-secure cloud storage public auditing scheme from lattice. We first formalize the identity-based data integrity audit scheme model which includes system framework and security model. Then, we present the concrete identity-based privacy-preserving and forward-secure cloud storage public auditing scheme from lattice which is provably secure under the hardness assumption of SIS problem. Furthermore, by utilizing encapsulation technology with random masking and the lattice basis delegation technique, our scheme can prevent malicious TPA from getting the knowledge of original data and can achieve forward security for resisting key exposure attack. Therefore, our protocol can achieve the goal that the user data is protected from being corrupted by the untrusted CSP and the privacy of user's data is secure against the malicious TPA under the quantum computer attack.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable comments and constructive suggestions to improve the quality of this article.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 598–609.

[2] S. K. Nayak and S. Tripathy, "SEPDP: Secure and efficient privacy preserving provable data possession in cloud storage," *IEEE Trans. Services Comput.*, early access, Mar. 29, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8327915, doi: 10.1109/TSC.2018.2820713.

[3] D. He, N. Kumar, H. Wang, L. Wang, and K.-K.-R. Choo, "Privacy-preserving certificateless provable data possession scheme for big data storage on cloud," *Appl. Math. Comput.*, vol. 314, pp. 31–43, Dec. 2017.

[4] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, "Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium," *J. Netw. Comput. Appl.*, vol. 82, pp. 56–64, 2017.

[5] W. Shen, G. Yang, J. Yu, H. Zhang, F. Kong, and R. Hao, "Remote data possession checking with privacy-preserving authenticators for cloud storage," *Future Gener. Comput. Syst.*, vol. 76, pp. 136–145, Nov. 2017.

[6] R. Ding, Y. Xu, J. Cui, and H. Zhong, "A public auditing protocol for cloud storage system with intrusion-resilience," *IEEE Syst. J.*, vol. 14, no. 1, pp. 633–644, Mar. 2020, doi: 10.1109/JSYST.2019.2923238.

[7] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Syst. J.*, vol. 12, no. 1, pp. 64–73, Mar. 2018.

[8] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1362–1375, Jun. 2016.

[9] J. Li, H. Yan, and Y. Zhang, "Efficient identity-based provable multi-copy data possession in multi-cloud storage," *IEEE Trans. Cloud Comput.*, early access, Jul. 16, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8764408, doi: 10.1109/TCC.2019.2929045.

[10] D. He, N. Kumar, S. Zeadally, and H. Wang, "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] H. Yan, J. Li, and Y. Zhang, "Remote data checking with a designated verifier in cloud storage," *IEEE Syst. J.*, early access, Jun. 4, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8730496, doi: 10.1109/JSYST.2019.2918022.

[12] J. Li, H. Yan, and Y. Zhang, "Certificateless public integrity checking of group shared data on cloud storage," *IEEE Trans. Services Comput.*, early access, Jan. 8, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8249859, doi: 10.1109/TSC.2018.2789893.

[13] H. Yan, J. Li, J. Han, and Y. Zhang, "A novel efficient remote data possession checking protocol in cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 78–88, Jan. 2017.

[14] C. Wang, S. S. 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. 2011.

[15] A. Ometov, A. Levina, P. Borisenko, R. Mostovoy, A. Orsino, and S. Andreev, "Mobile social networking under side-channel attacks: Practical security challenges," *IEEE Access*, vol. 5, pp. 2591–2601, 2017.

[16] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, "Lest we remember: Cold-boot attacks on encryption keys," *Commun. ACM*, vol. 52, no. 5, pp. 91–98, May 2009.

[17] M. Bellare and S. K. Miner, "A forward-secure digital signature scheme," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 1999, pp. 431–448. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-48405-1_28

[18] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1167–1179, Jun. 2015.

[19] J. Yu and H. Wang, "Strong key-exposure resilient auditing for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1931–1940, Aug. 2017.

[20] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute based encryption with continuous leakage-resilience," *Inf. Sci.*, vol. 484, pp. 113–134, May 2019.

[21] J. Li, Q. Yu, Y. Zhang, and J. Shen, "Key-policy attribute-based encryption against continual auxiliary input leakage," *Inf. Sci.*, vol. 470, pp. 175–188, Jan. 2019.

[22] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1984, pp. 47–53. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-39568-7_5

[23] H. Wang, J. Domingo-Ferrer, B. Qin, and Q. Wu, "Identity-based remote data possession checking in public clouds," *IET Inf. Secur.*, vol. 8, no. 2, pp. 114–121, Mar. 2014.

[24] H. Wang, "Identity-based distributed provable data possession in multi-cloud storage," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 328–340, Mar. 2015.

[25] S. Peng, F. Zhou, J. Xu, and Z. Xu, "Comments on 'identity-based distributed provable data possession in multicloud storage,'" *IEEE Trans. Services Comput.*, vol. 9, no. 6, pp. 996–998, Nov./Dec. 2016.

[26] C. Lan, H. Li, and W. Caifen, "Analysis of the comments on 'identity-based distributed provable data possession in multicloud storage,'" *IEEE Trans. Services Comput.*, early access, Nov. 29, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8123881, doi: 10.1109/TSC.2017.2778250.

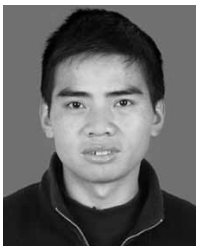
[27] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.

[28] M. Ajtai, "Generating hard instances of lattice problems," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 99–108.

[29] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 14th Annu. ACM Symp. Theory Comput. (STOC)*, 2008, pp. 197–206.
[30] D. Cash, D. Hofheinz, and E. Kiltz, "How to delegate a lattice basis," IACR Cryptol. ePrint Arch., Tech. Rep. 2009/351, 2009, p. 351. [Online]. Available: https://eprint.iacr.org/2009/351
[31] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2010, pp. 523–552. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-13190-5_27
[32] S. Agrawal, D. Boneh, and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2010, pp. 98–115. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-14623-7_6
[33] X. Zhang, H. Wang, and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," *Inf. Sci.*, vol. 472, pp. 223–234, Jan. 2019.
[34] W. Xu, D. Feng, and J. Liu, "Public verifiable proof of storage protocol from lattice assumption," in *Proc. IEEE Int. Conf. Intell. Control, Autom. Detection High-End Equip.*, Jul. 2012, pp. 133–137.
[35] T. Shuang, H. Li, C. Zhikun, and J. Yan, "A method of provable data integrity based on lattice in cloud storage," *J. Comput. Res. Develop.*, vol. 52, no. 8, pp. 1862–1872, 2015.
[36] Z. Liu, Y. Liao, X. Yang, Y. He, and K. Zhao, "Identity-based remote data integrity checking of cloud storage from lattices," in *Proc. 3rd Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Aug. 2017, pp. 128–135.
[37] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 169–178.
[38] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in *Proc. Annu. Int. Conf. theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2011, pp. 149–168. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-20465-4_10

HAIFENG LI received the B.S. degree in computer science from Hebei University and the M.S. degree in computer science from Northwest Normal University. He is currently pursuing the Ph.D. degree with the School of Software, Dalian University of Technology. His current research interests include applied cryptography, network security, cloud computing security, and big data security.

LIANGLIANG LIU received the Ph.D. degree in computer software and theory from the University of Chinese Academy of Sciences, Beijing, China, in 2014. He is currently a Lecturer with the School of Statistics and Information, Shanghai University of International Business and Economics. His main research interests are in information security, machine learning, natural language processing, and knowledge acquisition.



CAIHUI LAN received the Ph.D. degree in basic mathematics from the School of Mathematics and Statistics, Northwest Normal University, Lanzhou, China, in 2013. He is currently an Associate Professor with the School of Electronics and Information Engineering, Lanzhou City University. His main research interests include cryptography and information security, in particular, cryptographic protocols.

CAIFEN WANG received the Ph.D. degree in cryptography from the School of Communication Engineering, Xidian University, in 2003. She is currently a Professor with Shenzhen Technology University. Her main research interests include cryptography and information security, in particular, applied cryptography and security in cloud computing. She has been selected as a member of the Special Committee of Cryptography Algorithms and the Director of the China Cryptography Society.

HE GUO received the B.S. degree in computer science and technology from Jilin University, China, in 1982, and the M.S. degree in computer science and technology from the Dalian University of Technology, in 1988. He has been a Full Professor with the School of Software, Dalian University of Technology, since 2010. His research interests include computer vision, parallel and distributed computing, and cloud computing security.