Information Sciences 472 (2019) 223–234

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Identity-based key-exposure resilient cloud storage public auditing scheme from lattices

Xiaojun Zhang[a,b,c,*], Huaxiong Wang[b], Chunxiang Xu[c]

[a] School of Computer Science, Southwest Petroleum University, Chengdu 610500, China
[b] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
[c] Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

ARTICLE INFO

Article history:
Received 12 May 2018
Revised 17 August 2018
Accepted 9 September 2018
Available online 14 September 2018

Keywords:
Cloud storage auditing
Key-exposure resilient
Forward security
Lattice-assumptions
Post-quantum secure

ABSTRACT

With the rapid development of cloud auditing services, key exposure has been highlighted as a serious security issue. Using the exposed private key of a client, cloud servers can forge previous auditing proofs to cheat auditors. To date, a few pairing-based cloud storage auditing schemes addressing key exposure have been proposed. However, they are not secure from quantum attacks, and they rely on public key infrastructure (PKI), which involves complex certificate management. In this paper, we propose an efficient identity-based key-exposure resilient public auditing scheme from lattice assumptions in cloud storage. Our scheme is not only quantum-resistant, but eliminates the need to establish a PKI. We employ lattice basis delegation technique to update a client's private key flexibly, keeping the private key size constant. Based on the hardness of lattice assumptions, we prove the forward security of storage correctness guarantee against malicious cloud servers in detail, and that the proposed scheme preserves privacy against curious auditors. Furthermore, we conduct a performance comparison to demonstrate that our scheme is much more efficient and practical for post-quantum secure cloud storage.

1. Introduction

Nowadays, cloud computing platform provides both storage and computing services to clients, allowing them to access their data with high availability and low cost anywhere via the Internet without the need for complex local storage management [11]. Although clients enjoy the desirable benefits from cloud storage services, there are critical security challenges for data outsourcing [47]. The integrity of outsourced data has become a major concern for clients [40]. Unlike local storage devices, clients do not physically control the data stored in a remote cloud server. Thus, they are constantly worried about whether outsourced data are kept intact, especially if they are very important to clients.

Some approaches [3,18] have been proposed to address this issue of data integrity verification. However, they rely on the clients to check data integrity themselves, adding considerable burden of verification. Addressing this issue, public auditing has been introduced [29]. Public auditing employs a third-party auditor (TPA) to periodically check whether cloud servers have stored clients' data correctly. With this, clients are relieved from the task of data auditing. But it makes an impractical assumption, that the TPA have enough computation capability to bear expensive verification overhead. Moreover, based on

[*] Corresponding author.
E-mail address: zhangxjdzkd2012@163.com (X. Zhang).

https://doi.org/10.1016/j.ins.2018.09.013
0020-0255/© 2018 Elsevier Inc. All rights reserved.

the response auditing proofs from cloud servers, a curious TPA may derive primitive data of clients with powerful computing devices. Consequently, to maintain the privacy of clients' data against the curious TPA, some feasible technologies, e.g., random masking technique [29], have been exploited to prevent the TPA from breaking data privacy.

In the literature, the security of modern cryptographic systems applied in clouds wholly depends on the assumption that private keys are absolutely secure. This assumption may not be very accurate today with the proliferation in mobile devices which have limited key protection. Compared to large companies and organizations, most clients are less aware of security measures that safeguard their keys and may be more vulnerable to key exposure [24]. Worse still, to maintain a good reputation or save the storage space, cloud servers may hide the data loss incidents, or discard clients' data which are rarely accessed. In particular, with the exposed private key of a client, malicious cloud servers can tamper with the data and further forge response auditing proofs that are demanded by auditors to cheat them.

Although some public auditing schemes with novel properties have emerged [45,49], as a serious security issue, the key exposure for cloud storage auditing has not been a focus in previous research. Recently, Yu *et al.* [43] employed binary tree structure to update a client's private key, so that the forward security of the response auditing proofs, generated in time periods earlier than key exposure, can be guaranteed. Following that, Yu *et al.* [44] further proposed a strong key-exposure resilient auditing scheme to achieve the security of data auditing, not only earlier than but also later than key exposure. Despite the merits of key exposure resistance in these schemes, they impose large computation delay on the auditor due to the use of expensive bilinear pairings and modular exponentiations. In addition, they are built on the public key infrastructure (PKI). Therefore, they need substantial complex certificate management, especially in the distribution of public keys of clients in all the time periods, which might hinder the deployment of cloud storage auditing in practice. In contrast, identity-based cryptographic systems, first introduced by Shamir [26], can avoid the establishment of a PKI. In such a system, a key generation center (KGC) can generate the private key according to any recognized information of an identity, such as telephone number, address, or driving license number. Consequently, an identity-based key-exposure resilient cloud storage public auditing scheme has more advantages, especially in mobile cloud storage systems. In the literature [34,38], some cloud storage public auditing schemes possessing the advantages of identity-based systems have been proposed.

Nevertheless, future advances in quantum computers may render most conventional asymmetric cryptographic primitives unsafe [27], threatening the security of the aforementioned auditing schemes. Fortunately, lattice-based cryptography [25] is still secure against attacks by adversaries with quantum computers. It holds the best promise for post-quantum cryptography, as it enjoys very strong security proofs based on worst-case hardness, as well as efficient implementations.

To fill the gap, in this paper, we propose an efficient identity-based key-exposure resilient public auditing scheme from lattice assumptions in cloud storage, which is quantum-resistant. The contributions of this work are elaborated as follows.

- We integrate the forward secure signature [48] into the proposed scheme to achieve forward security. Specifically, we flexibly employ lattice basis delegation technique to update the private key of a client even after multiple time periods, which we call lazy update. Even if a malicious cloud server intercepts the client's private key in current time period, it is computationally infeasible to tamper with the client's data in previous time periods and further forge the response auditing proofs, that can pass the verification process successfully. Moreover, our scheme keeps each private key size constant in **KeyUpdate**, independent of the number of time periods.
- Our scheme is the first key-exposure resilient public auditing scheme which keeps either lattice-based or post-quantum secure settings. Based on the hardness assumption of ISIS problem, we prove the forward security of storage correctness guarantee against malicious cloud servers in detail. To ensure cloud storage data privacy, we exploit a preimage sampleable function (PSF) [15] to construct a random masking and integrate it into the proposed scheme, so that the curious TPA cannot derive the primitive data of clients by solving the linear equations.
- We conduct a performance comparison to demonstrate that the proposed scheme can dramatically reduce the computational costs on the TPA side. This is because the TPA only performs simple additions and multiplications over a moderate modulus, instead of time-consuming bilinear pairings and modular exponentiations. Besides, our scheme is designed on identity-based systems, eliminating the complicated certificate management for client's public keys in all the time periods. Therefore, our scheme is quite practical and applicable to the requirements of high efficiency in the post-quantum secure settings.

1.1. Related work

Cloud computing, with powerful storage and computing capabilities, has brought about great benefits to clients. Despite the merits of cloud computing, many security threats have emerged in clouds [12,32,50,54], which may undermine the confidentiality, reliability, and integrity of the outsourced data. Recently, some data encryption techniques have been proposed to guarantee data confidentiality, such as [13,14,19,21,22,36,39]. Some other cryptographic techniques, such as [8,9,20,46], can flexibly maintain data reliability. As one of the most important security issues for clients, data integrity needs to be addressed urgently, since any corrupt data may lead to serious consequences.

To cope with data integrity, Ateniese *et al.* [3] pioneered provable data possession (PDP) to guarantee that the data are correctly stored in an untrusted remote cloud server. In order to assure clients of both possession and retrievability of the outsourced data, Juels *et al.* [18] presented proof of retrievability (PoR). Subsequently, Shacham *et al.* [28] further proposed an improvement of PoR model with stateless verification, it is actually a private verification based on BLS signature technique.

Wang *et al.* [29] presented the first cloud storage public auditing scheme, which relies on the TPA to fulfil the auditing tasks on behalf of clients without downloading the entire data. With the rapid growth of cloud storage auditing services, auditing schemes supporting dynamic operations have emerged [6,35,41]. User revocation in shared cloud data auditing was also considered in [33]. The new indistinguishability obfuscation technique has been exploited to construct a public auditing scheme [49]. To address the issue of medical data integrity, a certificateless public auditing scheme for wireless body area networks has been proposed [17]. A light-weight pairing-free cloud storage auditing scheme for wireless medical sensor networks has also been given in [52]. To eliminate the complex certificate management in the PKI, some identity-based cloud storage auditing schemes have been proposed in the literature [30,34,38]. Particularly, in [30,34], clients can delegate their proxies to process massive data and outsource them to the cloud server. Moreover, to reduce the damage of the client's key exposure in cloud storage auditing, the pairing-based cloud storage auditing schemes with key-exposure resilience [42–44] have been proposed.

However, due to the security proof in [27], the conventional public key cryptographic algorithms will be broken by quantum computers, and hence the security of those auditing schemes mentioned above will be threatened. Lattice-based cryptography has been a promising technique for resisting quantum attacks, since it holds very strong security proofs based on worst-case hardness [25]. In 2008, Gentry *et al.* [15] introduced a preimage sampleable function, which is a basic tool to construct lattice-based signatures. Following that, some novel lattice-based signature schemes have been proposed [23,37,51]. More specifically, lattice-based linearly homomorphic signature schemes [5,10,31], fully homomorphic signature scheme [16] have also emerged. Some of them with homomorphic properties can be exploited to further construct lattice-based cloud storage auditing schemes. In recent, Zhang *et al.* [53] have given the detailed security analysis on the previous public proof of cloud storage from lattice, and have proposed the improved lattice-based auditing scheme.

2. Preliminaries

2.1. Lattices

Now we introduce the definitions of lattices as follows.

Definition 1. The matrix $B=\{b_1,\cdots,b_m\}\in\mathbb{R}^{m\times m}$ consists of m linearly independent vectors. The lattice m-dimension full-rank lattice is defined as $\Lambda=\{y\in\mathbb{R}^m:\exists z=(z_1,z_2,\cdots,z_m)^\top\in\mathbb{Z}^m, y=Bz=\sum_{i\in[m]}z_ib_i\}$.

Here $B=\{b_1,\cdots,b_m\}$ is a basis of the lattice Λ. Let $\widetilde{B}=\{\widetilde{b_1},\cdots,\widetilde{b_m}\}$ denote the Gram-Schmidt orthogonalization of the vectors b_1, $\cdots$, b_m taken in that order. We denote $\|B\|=\max_i\|b_i\|$, and refer to $\|\widetilde{B}\|$ as the Gram-Schmidt norm of B.

In this paper, our scheme is built on q-modular integer lattices defined by Ajtai [1].

Definition 2. For a matrix $A\in\mathbb{Z}_q^{n\times m}$, a vector $y\in\mathbb{Z}_q^n$, the integer lattices are defined as follows.

1. $\Lambda_q(A)=\{\varsigma\in\mathbb{Z}^m:\exists x\in\mathbb{Z}_q^n, \varsigma=A^\top x \mod q\}$.
2. $\Lambda_q^\perp(A)=\{e\in\mathbb{Z}^m: Ae=0 \mod q\}$.
3. $\Lambda_q^y(A)=\{e\in\mathbb{Z}^m: Ae=y \mod q\}$.

Definition 3. For any parameter $\delta>0$, define the Gaussian function on $\mathbb{R}^m$ centered at r as $\rho_{\delta,r}(x)=exp(-\pi\|x-r\|^2/\delta^2)$ and $\rho_{\delta,r}(L)=\sum_{x\in L}\rho_{\delta,r}(x)$, where L is a subset of $\mathbb{Z}^m$. The discrete Gaussian distribution over L with center r and parameter δ is $\forall x\in L$, $\mathcal{D}_{L,\delta,r}(x)=\rho_{\delta,r}(x)/\rho_{\delta,r}(L)$.

We define the hardness assumptions of lattice problems as follows.

Definition 4. The Inhomogeneous Small Integer Solution Problem (ISIS) is described below: Given a prime q, a matrix $A\in\mathbb{Z}_q^{n\times m}$, a vector $y\in\mathbb{Z}_q^n$, and a positive real number ζ, the goal is to solve a nonzero integer vector $e\in\mathbb{Z}^m$, such that $Ae=y$ mod q and $\|e\|\le\zeta$.

Similarly, the Small Integer Solution Problem (SIS) asks for a nonzero integer vector $e\in\mathbb{Z}^m$, such that $Ae=0$ mod q and $\|e\|\le\zeta$. As described in [15], for any poly-bounded $\zeta=poly(n)$ and for any prime $q>\zeta\cdot\omega(\sqrt{n\log n})$, the average-case SIS, ISIS problems are as hard as approximating the SIVP problem in the worst case to within certain factor $\zeta\cdot\tilde{O}(\sqrt{n})$.

The trapdoor generation algorithm and preimage sampleable function (PSF) are introduced as follows.

Lemma 1. *There exists a probabilistic polynomial-time algorithm* (PPT)TrapGen(q,n) [4] *that outputs* $(A\in\mathbb{Z}_q^{n\times m}, T_A\in\mathbb{Z}^{m\times m})$, *where* A *is statistically close to a uniform matrix in* $\mathbb{Z}_q^{n\times m}$, T_A *is a short lattice basis of* $\Lambda_q^\perp(A)$, *the Euclidean norm of all the rows is bounded by* $O(n\log n)$.

Lemma 2. *For any prime* q, *and the integer* $m\ge\lceil 2n\log q\rceil$, *the* PPT *algorithm* SamplePre(A,T_A,y,δ) [15] *takes as inputs a matrix* $A\in\mathbb{Z}_q^{n\times m}$, *a short lattice basis* $T_A\in\mathbb{Z}^{m\times m}$, *a vector* $y\in\mathbb{Z}_q^n$, *and a parameter* $\delta\ge\|\widetilde{T_A}\|\cdot\omega(\sqrt{\log m})$, *outputs a sample from a distribution that is statistically close to* $\mathcal{D}_{\Lambda_q^y(A),\delta}$, *where* $\mathcal{D}_{\Lambda_q^y(A),\delta}$ *is the discrete Gaussian distribution over* $\Lambda_q^y(A)$ *with parameter* δ.

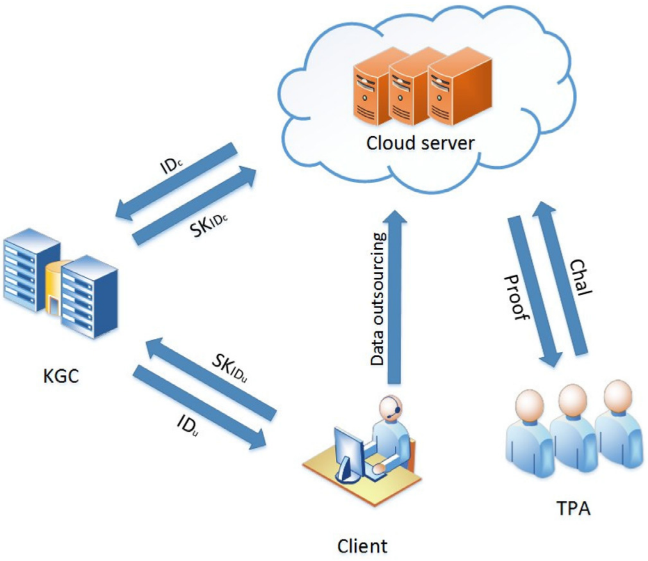

Fig. 1. System model of basic scheme.

Now we introduce the lattice delegation technique NewBasisDel in [2], which is important to update the private key of a client in each time period in our scheme. We first need to describe the distribution $\mathcal{D}_{m\times m}$ on matrices in $\mathbb{Z}_q^{m\times m}$, which is defined as $(\mathcal{D}_{\mathbb{Z}^m,\delta_R})^m$ conditioned on the resulting matrix R, where R is $\mathbb{Z}_q$-invertible, and $\sigma_R = \sqrt{n\log q}\cdot\omega(\sqrt{\log m})$.

Lemma 3. *The* PPT *algorithm* NewBasisDel(A, R, T_A, σ) *takes as inputs a rank n matrix* $A \in \mathbb{Z}_q^{n\times m}$, *a* $\mathbb{Z}_q$*-invertible matrix R sampled from the distribution* $\mathcal{D}_{m\times m}$, *a short lattice basis* T_A *of* $\Lambda_q^{\perp}(A)$, *and a parameter* $\sigma \geq \|\widetilde{T_A}\| \cdot \sigma_R\sqrt{m}\omega(\log^{3/2} m)$, *outputs a short lattice basis* T_B *of* $\Lambda_q^{\perp}(B)$, *where* $B = AR^{-1}$. NewBasisDel(A, R, T_A, σ) *performs in detail as follows.*

1. *Compute* $T_B' = \{Ra_1, \cdots, Ra_m\} \subseteq \mathbb{Z}^m$, *where* $T_A = \{a_1, \cdots, a_m\} \subseteq \mathbb{Z}^m$.
2. *Take as inputs* T_B' *and an arbitrary lattice basis of* $\Lambda_q^{\perp}(B)$, *output a lattice basis* T_B'' *of* $\Lambda_q^{\perp}(B)$, *such that its Gram-Schmidt norm is no more than that of* T_B'.
3. *Run the algorithm* RandBasis(T_B'', σ) *described in [7] to output a randomized lattice basis* T_B *of* $\Lambda_q^{\perp}(B)$.

In our provable forward security of storage correctness guarantee, we needs the PPT algorithm SampleR [2] to sample matrices in $\mathbb{Z}_q^{m\times m}$ from a distribution, which is statistically close to $\mathcal{D}_{m\times m}$. Moreover, the PPT algorithm SampleRwithBasis described in Lemma 4 is exploited to simulate a random short lattice basis.

Lemma 4. *The algorithm* SampleRwithBasis *[2] takes as inputs a prime* $q \geq 3$, $m \geq \lceil 2n\log q\rceil$, *a rank n matrix* $A \in \mathbb{Z}_q^{n\times m}$, *outputs a low-norm matrix R sampled from a distribution statistically close to* $\mathcal{D}_{m\times m}$, *and a random short lattice basis* T_B *of* $\Lambda_q^{\perp}(B)$, *where* $B = AR^{-1}$, *such that* $\|\widetilde{T_B}\| \leq \sigma_R/\omega(\sqrt{\log m})$ *with overwhelming probability.*

2.2. Basic system model and security definition

The basic system model of an identity-based cloud storage auditing scheme is shown in Fig. 1. It consists of four different entities: a client, cloud server, the key generation center (KGC), and the third-party auditor (TPA).

Client: As a cloud user, a client generates massive data, outsources them and corresponding authenticators to a cloud server for maintenance and management. The client deletes these data from local storage space.

Cloud server: Under the control of the cloud service provider (CSP), the cloud server provides facilitative storage space and powerful computation resources. Through it, clients can flexibly create, store, update data and request for retrievability.

KGC: Which is in charge of public-private key pair distribution of any identity, e.g., clients or cloud server.

TPA: Which can periodically check the integrity of the data stored in the remote cloud server on behalf of clients.

In our system model, clients will update their private keys at the end of each time period. We assume that the key exposure of clients occurs anywhere, and any adversary, including the cloud server, can capture the private keys of clients for cloud storage auditing in any time period.

The syntax of an identity-based key-exposure resilient cloud storage public auditing scheme consists of the following six polynomial-time algorithms.

Setup: This PPT algorithm takes as inputs a secure parameter κ, and the total number of time periods γ, outputs master public key PK and master secret key SK.

KeyExtract: This PPT algorithm takes as inputs master public-secret key pair $(\mathsf{PK}, \mathsf{SK})$, $ID_u = id_u \| \gamma$, here id_u denotes the identifier information of a client, outputs corresponding initial private key $\mathsf{SK}_{ID_u\|1}$, and sends it to id_u via a secure channel.

KeyUpdate: This PPT algorithm takes as inputs current time period i, the private key $\mathsf{SK}_{ID_u\|\tau}$ in previous time period τ, outputs the private key $\mathsf{SK}_{ID_u\|i}$ in current time period i, where $\tau < i$.

AuthGen: This PPT algorithm takes as inputs current time period i, the public-private key pair $(\mathsf{PK}_{ID_u\|i}, \mathsf{SK}_{ID_u\|i})$ of a client, and the data file $\mathcal{F}$, outputs the set of authenticators Ψ_i. The data file tag ξ_i is also generated in this phase. Finally, the client outsources $\{i, \mathcal{F}, \Psi_i, \xi_i\}$ to the cloud server, and deletes them in the local storage.

ProofGen: This PPT algorithm takes as inputs time period i, the data file $\mathcal{F}$, corresponding set of authenticators Ψ_i, and the challenge message *chal* from the TPA, outputs a response auditing proof Proof.

ProofVerify: This deterministic polynomial-time algorithm takes as inputs time period i, the public key $\mathsf{PK}_{ID_u\|i}$, the challenge message *chal*, and Proof from the cloud server, outputs *true* if the integrity of the data file is verified as correct. Otherwise, it outputs *false*.

Similar to the security definition in [43], now we describe an adversary $\mathcal{A}$ breaking storage correctness guarantee of an identity-based cloud storage public auditing scheme supporting forward security, by interacting with the challenger $\mathcal{B}$ in the following phases.

Setup: $\mathcal{B}$ generates the system public parameters and returns them to $\mathcal{A}$.

Query phase: $\mathcal{A}$ can adaptively query as follows.

- Private key query: $\mathcal{A}$ can adaptively query for the private key of any ID_u in time period i, $\mathcal{B}$ generates corresponding private key $\mathsf{SK}_{ID_u\|i}$ and sends it to $\mathcal{A}$.
- AuthGen query: $\mathcal{A}$ can adaptively query the authenticators of a series of data blocks F_1, F_2, $\cdots$, F_ℓ to $\mathcal{B}$ in time period i. $\mathcal{B}$ computes corresponding set of authenticators for $F_i (i = 1, \cdots, \ell)$, and sends them back to $\mathcal{A}$. $\mathcal{A}$ saves all data blocks $\mathcal{F} = \{F_1, F_2, \cdots, F_\ell\}$ and corresponding set of authenticators. $\mathcal{A}$ can choose to still stay in query phase or go to the break-in phase.

Break-in phase: $\mathcal{A}$ queries on the private key for a specific identity ID_u in the break-in time period τ, $\mathcal{B}$ provides $\mathcal{A}$ with the private key $\mathsf{SK}_{ID_u\|\tau}$ for that time period by the **KeyUpdate** algorithm.

Then $\mathcal{B}$ sends $\mathcal{A}$ a challenge message $chal = \{j, v_{i,j}\}_{j\in\mathcal{L}}$, where $\mathcal{L} = \{l_1, \cdots, l_c\}$, and a time period i^* $(i^* < \tau)$. It requires $\mathcal{A}$ to provide a response auditing proof of possession for $F_{l_1}, \cdots, F_{l_c}$ of data file $\mathcal{F} = \{F_1, F_2, \cdots, F_\ell\}$, under the challenge message *chal* in time period i^*.

Attack phase: After querying for polynomial times as before, $\mathcal{A}$ announces to $\mathcal{B}$ the identity ID_u^* which will be challenged. Then $\mathcal{A}$ forges a response auditing proof Proof^* for the data blocks indicated by *chal* in time period i^*. If $\mathsf{ProofVerify}(\mathsf{PK}_{ID_u^*\|i^*}, i^*, chal, \mathsf{Proof}^*) = 1$, then $\mathcal{A}$ wins the above game.

We claim that an identity-based cloud storage public auditing scheme achieves forward security if the adversary $\mathcal{A}$ in above game causes the challenger $\mathcal{B}$ to accept its forged response auditing proof in time period i^* prior to key exposure only with negligible probability.

3. Our identity-based key-exposure resilient cloud storage public auditing scheme

3.1. Lattice-based technique for forward security

In our construction, we need to set the pre-specified total number of time periods γ ahead of time. We take advantage of lattice basis delegation technique $\mathsf{NewBasisDel}$ to update private key in each time period flexibly. Unlike [48], in **KeyUpdate**, any client can evolve the private key directly from previous time period $\tau \le \gamma - 1$ to current time period $i(\tau < i \le \gamma)$. More specifically, the client initially runs $\mathsf{NewBasisDel}$ to evolve private key in each time period $i = 1, 2, \cdots, \gamma$, allowing him/her to generate a series of private keys $\mathsf{SK}_{ID_u\|1}, \mathsf{SK}_{ID_u\|2}, \cdots, \mathsf{SK}_{ID_u\|\gamma}$, where $ID_u = id_u\|\gamma$ and id_u denotes the identifier information of the client. According to ID_u, current time period i, and related hash function, the corresponding public key $\mathsf{PK}_{ID_u\|i}$ in each time period i can be computed by the client, cloud server, or TPA, thereby avoiding the complex certificate management in PKI. The client employs $\mathsf{SK}_{ID_u\|i}$ to compute authenticators of data blocks in time period i, and outsources them to the cloud server. Once the auditing tasks in time period i is received, the TPA computes $\mathsf{PK}_{ID_u\|i}$, further validates the authenticators, and makes sure these data blocks are kept intact. When the time period changes from τ to i, where $\tau < i$, the client deletes $\mathsf{SK}_{ID_u\|\tau}, \mathsf{SK}_{ID_u\|\tau+1} \cdots, \mathsf{SK}_{ID_u\|i-1}$ from his storage, the new private keys are evolved $\mathsf{SK}_{ID_u\|i}, \mathsf{SK}_{ID_u\|i+1}, \cdots, \mathsf{SK}_{ID_u\|\gamma}$. From this time period onwards, the cloud server cannot tamper with any data as well as corresponding authenticators that can be verified under $\mathsf{PK}_{ID_u\|\tau}$ in any previous time period $\tau < i$, even if the private keys $\mathsf{SK}_{ID_u\|i}, \mathsf{SK}_{ID_u\|\tau+1}, \cdots, \mathsf{SK}_{ID_u\|\gamma}$ are exposed.

3.2. The proposed scheme

Our identity-based key-exposure resilient cloud storage public auditing scheme from lattice consists of the following six polynomial-time algorithms.

Setup: The system initialization algorithm is composed of the following four steps.

1. The system first preprocesses a data file $\mathcal{F}$ into ℓ data blocks $\mathcal{F} = \{F_1, F_2, \cdots, F_\ell\}$, each $F_j \in \mathbb{Z}_q^m$, $1 \le j \le \ell$. Here the data file $\mathcal{F}$ is actually an $(m \times \ell)$-dimension matrix, the system also sets its unique identification index to be $index \in \{0, 1\}^*$.
2. The system sets the discrete Gaussian distribution χ. For each time period $i = 1, \cdots, \gamma$, to make sure SamplePre and NewBasisDel to execute properly, the system sets two series of security Gaussian parameters $\delta = (\delta_0, \delta_1, \cdots, \delta_\gamma)$, $\sigma = (\sigma_0, \sigma_1, \cdots, \sigma_\gamma)$, respectively.
3. The system sets four collision-resistant hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^{m\times m}$, the output value of H_1 is distributed in $\mathcal{D}_{m\times m}$, $H_2 : \mathbb{Z}_q^{n\times m} \times \{0,1\}^* \to \mathbb{Z}_q^n$, $H_3 : \{0,1\}^* \to \mathbb{Z}_q^n$, and $H_4 : \mathbb{Z}_q^n \to \mathbb{Z}_q$.
4. The KGC runs $\mathsf{TrapGen}(q, n)$ to generate master public-secret key pair $(A \in \mathbb{Z}_q^{n\times m}, T_A \in \mathbb{Z}_q^{m\times m})$. The client selects a lightweight signature algorithm SSig to ensure the integrity of the unique identification index, with the public-secret key pair being (*spk, ssk*).

Finally, the system initialization outputs the system public parameters $\Omega = (A, H_1, H_2, H_3, H_4, spk, \delta, \sigma)$.

KeyExtract: The KGC takes as inputs Ω, $ID_u = id_u \| \gamma$, and its master secret key T_A, outputs the initial public-private key pair of the client as follows.

1. Compute the initial public key $A_{ID_u\|1} = A(R_{ID_u\|1})^{-1} \in \mathbb{Z}_q^{n\times m}$, where $R_{ID_u\|1} = H_1(ID_u\|1)$.
2. Run $\mathsf{NewBasisDel}(A, R_{ID_u\|1}, T_A, \sigma_1)$ to generate a random short lattice basis $T_{ID_u\|1} \in \mathbb{Z}_q^{m\times m}$ of $\Lambda_q^\perp(A_{ID_u\|1})$ as the initial private key of the client.

The KGC sends $T_{ID_u\|1}$ to the client via a secure channel. In a similar manner, the KGC computes public key $A_c = A(R_{ID_c})^{-1} = A(H_1(ID_c))^{-1}$ of the cloud server ID_c, runs $\mathsf{NewBasisDel}(A, R_{ID_c}, T_A, \sigma_0)$ to generate T_c as the private key, and sends T_c to the cloud server via a secure channel.

KeyUpdate: Taking as inputs Ω, current time period *i*, and the private key $T_{ID_u\|\tau}$ of the client in previous time period τ, where $\tau < i < \gamma$, the client generates the private key $T_{ID_u\|i}$ as follows.

1. Compute the public key $A_{ID_u\|\tau} = A(R_{ID_u\|\tau})^{-1} \in \mathbb{Z}_q^{n\times m}$ in time period τ, where $R_{ID_u\|\tau} = H_1(ID_u\|\tau) \cdots H_1(ID_u\|1) \in \mathbb{Z}_q^{m\times m}$.
2. Compute $R_{ID_u\|\tau\to i} = H_1(ID_u\|i) \cdots H_1(ID_u\|\tau + 1)$, run the algorithm $\mathsf{NewBasisDel}(A_{ID_u\|\tau}, R_{ID_u\|\tau\to i}, T_{ID_u\|\tau}, \sigma_i)$ to generate a random short lattice basis $T_{ID_u\|i} \in \mathbb{Z}_q^{m\times m}$ of $\Lambda_q^\perp(A_{ID_u\|i})$, where the public key $A_{ID_u\|i}$ in time period *i* is $A_{ID_u\|i} = A_{ID_u\|\tau}(R_{ID_u\|\tau\to i})^{-1} = A(R_{ID_u\|i})^{-1} \in \mathbb{Z}_q^{n\times m}$.

AuthGen: For each data block $F_j \in \mathbb{Z}_q^m$ of a data file $\mathcal{F} = \{F_1, F_2, \cdots, F_\ell\}$, $1 \le j \le \ell$, with its unique identification index being $index \in \{0, 1\}^*$, the client takes as inputs current time period *i*, public-private key pair $(A_{ID_u\|i}, T_{ID_u\|i})$ of the client, and the public key A_c of the cloud server, outputs corresponding authenticator of F_j as follows.

1. Compute *n* vectors $\lambda_{i,k} = H_2(A_{ID_u\|i}\|index\|k) \in \mathbb{Z}_q^n$, $k = 1, \cdots, n$.
2. Compute $\rho_j = H_3(index\|j) + A_c F_j$, the inner products $h_{i,j,k} = \langle \rho_j, \lambda_{i,k} \rangle$, $1 \le j \le \ell$, $1 \le k \le n$, and parse $h_{i,j} = (h_{i,j,1}, \cdots, h_{i,j,n})^\top \in \mathbb{Z}_q^n$.
3. Run $\mathsf{SamplePre}(A_{ID_u\|i}, T_{ID_u\|i}, h_{i,j}, \delta_i)$ to generate the authenticator $\theta_{i,j}$ in time period *i*.

Denote the set of authenticators by $\Psi_i = \{\theta_{i,j}\}_{1\le j\le \ell}$, and compute $\xi_i = index\|i\|\mathsf{SSig}_{ssk}(index\|i)$ as the data file tag of $\mathcal{F}$. Finally, the client outsources $\{i, \mathcal{F}, \Psi_i, \xi_i\}$ to the cloud server and deletes them in the local storage.

ProofGen: The TPA first verifies the integrity of the data file tag ξ_i by checking whether $\mathsf{SSig}_{ssk}(index\|i)$ is a valid signature under the public key *spk*. The TPA aborts if the verification process fails. Otherwise, the TPA recovers *index*‖*i*, and further generates the following auditing challenge message.

1. Select a random *c*-element subset $\mathcal{L} = \{l_1, \cdots, l_c\}$ of the universal set {1, 2, ⋯, ℓ}, locating the data blocks needed to be checked.
2. Select a random binary string $\nu_i = \{\nu_{i,l_1}, \nu_{i,l_2}, \cdots, \nu_{i,l_c}\} \in \{0,1\}^c$, and send the challenge message $chal = \{j, \nu_{i,j}\}_{j\in\mathcal{L}}$ to the cloud server.

Once receiving $chal = \{j, \nu_{i,j}\}_{j\in\mathcal{L}}$, the cloud server computes the aggregate authenticator $\theta_i = \sum_{j\in\mathcal{L}} \nu_{i,j}\theta_{i,j}$, and the combined message $\mu_i' = \sum_{j\in\mathcal{L}} \nu_{i,j} F_j$. To prevent the TPA from deriving the primitive data blocks of the client, the cloud server needs to further blind μ_i'. It chooses a random vector $w_i \leftarrow \mathbb{Z}_q^n$, runs $\mathsf{SamplePre}(A_c, T_c, w_i, \delta_0)$ to generate the signature β_i of w_i. Finally, the cloud server computes $\mu_i = \beta_i + H_4(w_i)\mu_i'$, and sends $\mathsf{Proof} = \{i, \mu_i, \theta_i, w_i\}$ as the response auditing proof to the TPA.

ProofVerify: Once receiving $\mathsf{Proof} = \{i, \mu_i, \theta_i, w_i\}$, the TPA proceeds to check the validity of the response auditing proof as follows.

1. Compute *n* vectors $\lambda_{i,k} = H_2(A_{ID_u\|i}\|index\|k) \in \mathbb{Z}_q^n$, $k = 1, \cdots, n$.
2. Compute $\eta_i = H_4(w_i)\sum_{j\in\mathcal{L}} \nu_{i,j} H_3(index\|j) + A_c\mu_i - w_i \in \mathbb{Z}_q^n$.
3. Compute the inner products $h_{\mathcal{L}k} = \langle \eta_i, \lambda_{i,k} \rangle$, $k = 1, \cdots, n$ and parse $h_{\mathcal{L}} = (h_{\mathcal{L}1}, h_{\mathcal{L}2}, \cdots, h_{\mathcal{L}n})^\top \in \mathbb{Z}_q^n$.
4. Check the verification equation $H_4(w_i)A_{ID_u\|i}\theta_i = h_{\mathcal{L}}$ and $0 < \|\theta_i\| \le c\delta_i\sqrt{m}$ whether or not hold.

4. Evaluation of the proposed scheme

4.1. Correctness

The correctness of $\mathsf{Proof} = \{i, \mu_i, \theta_i, w_i\}$ in the verification equation is elaborated as follows.

$$\begin{aligned}
H_4(w_i)A_{ID_u\|i}\theta_i &= H_4(w_i)A_{ID_u\|i}\sum_{j\in\mathcal{L}}\nu_{i,j}\theta_{i,j}\\
&= H_4(w_i)\sum_{j\in\mathcal{L}}\nu_{i,j}A_{ID_u\|i}\theta_{i,j}\\
&= H_4(w_i)\sum_{j\in\mathcal{L}}\nu_{i,j}h_{i,j}\\
&= H_4(w_i)\sum_{j\in\mathcal{L}}\nu_{i,j}(\langle\rho_j,\lambda_{i,1}\rangle,\langle\rho_j,\lambda_{i,2}\rangle,\cdots,\langle\rho_j,\lambda_{i,n}\rangle)^\top\\
&= H_4(w_i)(\langle\sum_{j\in\mathcal{L}}\nu_{i,j}\rho_j,\lambda_{i,1}\rangle,\cdots,\langle\sum_{j\in\mathcal{L}}\nu_{i,j}\rho_j,\lambda_{i,n}\rangle)^\top\\
&= H_4(w_i)(\langle\sum_{j\in\mathcal{L}}\nu_{i,j}H_3(index\|j)+A_c\mu_i',\lambda_{i,1}\rangle,\\
&\quad\cdots,\langle\sum_{j\in\mathcal{L}}\nu_{i,j}H_3(index\|j)+A_c\mu_i',\lambda_{i,n}\rangle)^\top\\
&= (\langle H_4(w_i)\sum_{j\in\mathcal{L}}\nu_{i,j}H_3(index\|j)+A_c(\mu_i-\beta_i),\lambda_{i,1}\rangle,\\
&\quad\cdots,\langle H_4(w_i)\sum_{j\in\mathcal{L}}\nu_{i,j}H_3(index\|j)+A_c(\mu_i-\beta_i),\lambda_{i,n}\rangle)^\top\\
&= (\langle H_4(w_i)\sum_{j\in\mathcal{L}}\nu_{i,j}H_3(index\|j)+A_c\mu_i-w_i,\lambda_{i,1}\rangle,\\
&\quad\cdots,\langle H_4(w_i)\sum_{j\in\mathcal{L}}\nu_{i,j}H_3(index\|j)+A_c\mu_i-w_i,\lambda_{i,n}\rangle)^\top\\
&= (\langle\eta_i,\lambda_{i,1}\rangle,\cdots,\langle\eta_i,\lambda_{i,n}\rangle)^\top\\
&= (h_{\mathcal{L}1},h_{\mathcal{L}2},\cdots,h_{\mathcal{L}n})^\top\\
&= h_{\mathcal{L}}
\end{aligned}$$

Thus, the verification equation $H_4(w_i)A_{ID_u\|i}\theta_i = h_{\mathcal{L}}$ holds. On the other hand, since $\theta_{i,\,j}$ is an authenticator of a data block F_j in the time period i, for each $j \in \mathcal{L}$, $0 < \|\theta_{i,j}\| \le \delta_i\sqrt{m}$, thereby $0 < \|\theta_i\| \le c\delta_i\sqrt{m}$ also holds.

4.2. Provable security of the proposed scheme

Now, we provide the provable security of our scheme. Theorem 1 demonstrates the malicious cloud server as an adversary can never cheat the TPA to pass the verification process, by forging the response auditing proof of previous time period, even if the private key in current time period is exposed. Theorem 2 demonstrates that the proposed scheme preserves privacy against the curious TPA, so that the curious TPA cannot derive any data block of the data file from the response auditing proof.

Theorem 1. *The proposed scheme achieves key exposure resistance, provided that the hardness assumption of ISIS problem holds.*

Proof. We first assume that there exists an adversary $\mathcal{A}$ (malicious cloud server) breaking forward security of the proposed scheme with non-negligible probability ϵ. Then, we will show how to construct a challenger $\mathcal{B}$ solving the hardness assumption of ISIS problem with also non-negligible probability ϵ', by running the adversary $\mathcal{A}$ as a subroutine. Here $\mathcal{B}$ plays the role of a client or the TPA. Initially, we assume that:

- For each time period $i = 1, \cdots, \gamma$, $\mathcal{A}$ can perform polynomial numbers of queries H_1 on any identity adaptively.
- No matter when $\mathcal{A}$ performs the H_1-query on an identity in time period i, we assume that it has queried in time period $\tau < i$.
- No matter when $\mathcal{A}$ submits a client's private key query, we assume that it has performed all relevant H_1 queries before.

First of all, $\mathcal{B}$ randomly guesses $\imath = i^*$ $(1 \le i^* \le \gamma)$ as the time period when $\mathcal{A}$ forges the response auditing proof to pass the verification process. We suppose that $\mathcal{B}$ is given an instance of ISIS problem $(P, \hbar) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$ and manages to solve the vector $\theta_{\imath,t}^* \in \mathbb{Z}_q^m$ such that $P\theta_{\imath,t}^* = \hbar$ and $0 < \|\theta_{\imath,t}^*\| \le \delta_\imath\sqrt{m}$. $\mathcal{B}$ maintains five lists L_1, L_2, L_3, L_4, L_5, which are initialized to be empty. Let Q be the maximum number of H_1 queries, $\mathcal{B}$ independently chooses $\imath$ random numbers $Q_1^*, Q_2^*, \cdots, Q_\imath^* \in [Q] = \{1, 2, \cdots, Q\}$, and generates $\imath$ matrices $R_1^*, \cdots, R_\imath^* \leftarrow \mathcal{D}_{m\times m}$ by running $\mathsf{SampleR}$. Finally, $\mathcal{B}$ sets $A = PR_\imath^* \cdots R_1^*$, returns the system public parameters $\Omega = (A, H_1, H_2, H_3, H_4, spk, \delta, \sigma)$ to $\mathcal{A}$. $\mathcal{A}$ performs the following.

H_1 query: For the query on $ID_u\|i$, if $1 \le i \le i^* = \imath$, $\mathcal{B}$ answers such Q_{H_1}-th query as follows. If $Q_{H_1} = Q_i^*$, $\mathcal{B}$ sets $H_1(ID_u\|i) = R_i^*$, and returns R_i^* to $\mathcal{A}$. Otherwise, if $Q_{H_1} \neq Q_i^*$, $\mathcal{B}$ computes $A_{ID_u\|i-1} = A(R_{i-1}^* \cdots R_1^*)^{-1}$, runs $\mathsf{SampleRwithBasis}(A_{ID_u\|i-1})$ to generate $R_i \leftarrow \mathcal{D}_{m\times m}$, and a random short lattice basis $T_{ID_u\|i}$ for $A_{ID_u\|i} = A_{ID_u\|i-1}(R_i)^{-1}$. Finally, $\mathcal{B}$ adds $(ID_u\|i, A_{ID_u\|i}, R_i, T_{ID_u\|i})$ to list L_1, returns R_i to $\mathcal{A}$.

If $i = \imath + 1$, $\mathcal{B}$ runs $\mathsf{SampleRwithBasis}(A_{ID_u\|\imath})$ to generate $R_{\imath+1} \leftarrow \mathcal{D}_{m\times m}$, and a random short lattice basis $T_{ID_u\|\imath+1}$ for $A_{ID_u\|\imath+1} = A_{ID_u\|\imath}(R_{\imath+1})^{-1}$. Finally, $\mathcal{B}$ adds $(ID_u\|\imath+1, A_{ID_u\|\imath+1}, R_{\imath+1}, T_{ID_u\|\imath+1})$ to list L_1, returns $R_{\imath+1}$ to $\mathcal{A}$.

If $i > \imath + 1$, $\mathcal{B}$ performs as follows. As we aforementioned earlier, $\mathcal{B}$ can search $(ID_u\|i-1, A_{ID_u\|i-1}, R_{i-1}, T_{ID_u\|i-1})$ in list L_1, $\mathcal{B}$ chooses a matrix $R_i \leftarrow \mathcal{D}_{m\times m}$, and runs $\mathsf{NewBasisDel}(A_{ID_u\|i-1}, R_i, T_{ID_u\|i-1}, \sigma_i)$ to generate a random short lattice basis $T_{ID_u\|i}$ for $A_{ID_u\|i} = A_{ID_u\|i-1}(R_i)^{-1}$. Finally, $\mathcal{B}$ adds $(ID_u\|i, A_{ID_u\|i}, R_i, T_{ID_u\|i})$ to L_1, and returns R_i to $\mathcal{A}$.

H_2 query: Upon receiving $\mathcal{A}'s$ query on a distinct $(A_{ID_u\|i}, index, k)$, $\mathcal{B}$ first checks if the H_2 value of $(A_{ID_u\|i}, index, k)$ exists in list L_2. If it does, the previously defined value is returned. Otherwise, $\mathcal{B}$ randomly chooses a vector from $\mathbb{Z}_q^n$, adds it to L_2, and returns it to $\mathcal{A}$.

Similarly, once $\mathcal{A}$ queries to H_3 on (*index, j*), and $\mathcal{A}$ queries to H_4 on w_i in time period i, $\mathcal{B}$ checks if the H_3 value of (*index, j*) and the H_4 value of w_i exist in lists L_3, L_4, respectively. If so, the previously defined value are returned, respectively. Otherwise, $\mathcal{B}$ chooses a random vector from $\mathbb{Z}_q^n$, a random number from $\mathbb{Z}_q$, adds them to L_3 and L_4, respectively, and returns them to $\mathcal{A}$.

Private key query: Once $\mathcal{A}$ queries for the private key of ID_u in time period i, $\mathcal{B}$ provides $\mathcal{A}$ with corresponding private key $T_{ID_u\|i}$ as follows.

For the previous time period j-th query, let $j \in [\gamma]$ be the smallest time period when $H_1(ID_u\|j) \neq R_j^*$. Since we have assumed that $\mathcal{A}$ have made H_1 query on $ID_u\|j$ as before, we retrieve the saved tuple $(ID_u\|j, A_{ID_u\|j}, R_j, T_{ID_u\|j})$ from list L_1. We can also construct the matrix $A_{ID_u\|j} = A(R_1^*)^{-1} \cdots (R_{j-1}^*)^{-1}(H_1(ID_u\|j))^{-1}$ and $T_{ID_u\|j}$ is a random short lattice basis for $A_{ID_u\|j}$. As $\mathcal{B}$ can compute the low norm matrix $R_{ID_u\|j\to i} = H_1(ID_u\|i) \cdots H_1(ID_u\|j+1)$ as before, then $\mathcal{B}$ runs $\mathsf{NewBasisDel}(A_{ID_u\|j}, R_{ID_u\|j\to i}, T_{ID_u\|j}, \sigma_i)$ to generate the public key $A_{ID_u\|i} = A_{ID_u\|j}(R_{ID_u\|j\to i})^{-1}$, and a random short lattice basis $T_{ID_u\|i}$ as the private key in time period i. Finally, $\mathcal{B}$ adds $(ID_u\|i, A_{ID_u\|i}, T_{ID_u\|i})$ to list L_5, and returns $T_{ID_u\|i}$ to $\mathcal{A}$.

AuthGen query: $\mathcal{A}$ adaptively selects a data file $\mathcal{F}' = \{F_1', F_2', \cdots, F_\ell'\}$, with its identification index being $index' \in \{0,\ 1\}^*$, and the identity ID_u, then $\mathcal{A}$ queries on corresponding authenticators in time period i to $\mathcal{B}$. $\mathcal{B}$ performs as follows.

1. $\mathcal{B}$ generates $T_{ID_u\|i}$ in the same way of answering the query on the private key in time period i.
2. $\mathcal{B}$ looks into list L_2 to get $(A_{ID_u\|i}, index', k, \lambda_{i,k}')$, $k = 1, \cdots, n$, and looks into list L_3 to get $(index',\ j,\ H_3(index'\|j))$, $j = 1, \cdots, \ell$. Then $\mathcal{B}$ computes each $\rho_j' = H_3(index'\|j) + A_cF_j'$, $j = 1, \cdots, \ell$, and computes $h_{i,j}' = (h_{i,j,1}', \cdots, h_{i,j,n}')^\top \in \mathbb{Z}_q^n$, where $h_{i,j,k}' = \langle \rho_j', \lambda_{i,k}' \rangle$, $1 \le j \le \ell$, $1 \le k \le n$. For each $j = 1, \cdots, \ell$, $\mathcal{B}$ runs $\mathsf{SamplePre}(A_{ID_u\|i}, T_{ID_u\|i}, h_{i,j}', \delta_i)$ to generate the authenticator $\theta_{i,j}'$. Thereby $\mathcal{B}$ can generate the set of authenticators $\Psi' = \{\theta_{i,1}', \cdots, \theta_{i,\ell}'\}$ in the repeated steps.
3. $\mathcal{B}$ takes advantage of SSig to compute $\xi_i' = index'\|i\|\mathsf{SSig}_{ssk}(index'\|i)$.

Finally, $\mathcal{B}$ returns $\{i, \mathcal{F}', \xi_i', \Psi'\}$ to $\mathcal{A}$.

Break-in phase: Once $\mathcal{A}$ queries on the private key for a specific identity ID_u in the break-in time period τ, $\mathcal{B}$ looks into list L_5 to find corresponding private key $T_{ID_u\|\tau}$ and sends it to $\mathcal{A}$. Then $\mathcal{B}$ sends $\mathcal{A}$ a challenge message $chal = \{j, \nu_{\imath,j}\}_{j\in\mathcal{L}}$ and a time period $\imath = i^*$ $(\imath < \tau)$. $\mathcal{B}$ requests $\mathcal{A}$ to provide the response auditing proof of possessing data blocks $F_{l_1}, \cdots, F_{l_c}$ of $\mathcal{F} = \{F_1, F_2, \cdots, F_\ell\}$, under the challenge message *chal* in time period $\imath = i^*$.

Attack phase: After polynomially many queries as before, $\mathcal{A}$ announces to $\mathcal{B}$ the identity ID_u^* which will be challenged. Once $\mathcal{B}$ sends the challenge message $chal = \{j, \nu_{\imath,j}\}_{j\in\mathcal{L}}$ to $\mathcal{A}$ in time period $\imath = i^*$. $\mathcal{A}$ performs as follows.

The adversary $\mathcal{A}$, in the role of the malicious cloud server, may try to forge the response auditing proof in time period $\imath = i^*$. More specifically, $\mathcal{A}$ tampers with the client's data F_t as F_t^*, and forges corresponding authenticator $\theta_{\jmath,\,t}$ as $\theta_{\imath,t}^*$ with non-negligible probability ϵ, then $\mathcal{A}$ performs as follows.

1. Choose a random vector $w_\imath \leftarrow \mathbb{Z}_q^n$, run $\mathsf{SamplePre}(A_c, T_c, w_\imath, \delta_0)$ to generate the signature $\beta_\imath$ of $w_\imath$, compute $\mu_\imath^* = \beta_\imath + H_4(w_\imath)(\sum_{j\in\mathcal{L}, j\neq t} \nu_{\imath,j}F_j + \nu_{\imath,t}F_t^*)$.
2. Compute the aggregate authenticator $\theta_\imath^* = \sum_{j\in\mathcal{L}, j\neq t} \nu_{\imath,j}\theta_{\imath,j} + \nu_{\imath,t}\theta_{\imath,t}^*$.

Finally, $\mathcal{A}$ sends the forged response auditing proof $\mathsf{Proof}^* = \{\imath, \mu_\imath^*, \theta_\imath^*, w_\imath\}$ to $\mathcal{B}$, which can successfully pass the verification equation:

$$\begin{aligned} H_4(w_\imath)A_{ID_u^*\|\imath}\theta_\imath^* = {} & (\langle H_4(w_\imath)\sum_{j\in\mathcal{J}} \nu_{\imath,j}H_3(index\|j) + A_c\mu_\imath^* - w_i, \lambda_{\imath,1}\rangle, \cdots, \\ & \langle H_4(w_\imath)\sum_{j\in\mathcal{J}} \nu_{\imath,j}H_3(index\|j) + A_c\mu_\imath^* - w_\imath, \lambda_{\imath,n}\rangle)^\top. \end{aligned}$$

For simplicity, set G to be a matrix such that the k-th row of G is the vector $\lambda_{\iota,k} = H_2(A_{ID_u^*\|\iota}\|index\|k) \in \mathbb{Z}_q^n$ for $1 \le k \le n$. Thus:

$$H_4(w_\iota)A_{ID_u^*\|\iota}\theta_\iota^* = G(H_4(w_\iota)\sum_{j\in\mathcal{J}}\nu_{\iota,j}H_3(index\|j) + A_c\mu_\iota^* - w_\iota).$$

As a matter of fact, $H_4(w_\iota)A_{ID_u^*\|\iota}\theta_\iota^* = H_4(w_\iota)A_{ID_u^*\|\iota}(\sum_{j\in\mathcal{L},j\neq t}\nu_{\iota,j}\theta_{\iota,j} + \nu_{\iota,t}\theta_{\iota,t}^*)$.

With respect to the combined message $\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}F_j$ and aggregate authenticator $\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}\theta_{\iota,j}$, we can reconsider that the cloud server can generate the valid response auditing proof according to the challenge message $chal = \{j, \nu_{\iota,j}\}_{j\in\mathcal{J},j\neq t}$ from $\mathcal{B}$ as follows. $\mathcal{A}$ uniformly chooses $w_{\iota 1} \leftarrow \mathbb{Z}_q^n$ as another random vector, runs $\mathsf{SamplePre}(A_c, T_c, w_{\iota 1}, \delta_0)$ to generate β_{11}. Then, $\mathcal{A}$ computes $\mu_{\iota 1} = \beta_{\iota 1} + H_4(w_{\iota 1})\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}F_j$, and $\theta_{\iota 1} = \sum_{j\in\mathcal{L},j\neq t}\nu_{\iota,j}\theta_{\iota,j}$. Thus, $\mathcal{A}$ can generate the valid response auditing proof $\mathsf{Proof}_1 = \{\iota, \mu_{\iota 1}, \theta_{\iota 1}, w_{\iota 1}\}$. Hence, the following verification equation holds:

$$\begin{aligned}
H_4(w_{\iota 1})A_{ID_u^*\|\iota}\theta_{\iota 1} &= G(H_4(w_{\iota 1})\sum_{j\in\mathcal{L},j\neq t}\nu_{\iota,j}H_3(index\|j) + A_c\mu_{\iota 1} - w_{\iota 1})\\
&= G(H_4(w_{\iota 1})\sum_{j\in\mathcal{L},j\neq t}\nu_{\iota,j}H_3(index\|j)\\
&\quad + A_c(\beta_{\iota 1} + H_4(w_{\iota 1})\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}F_j) - w_{\iota 1})\\
&= G(H_4(w_{\iota 1})\sum_{j\in\mathcal{L},j\neq t}\nu_{\iota,j}H_3(index\|j)\\
&\quad + A_cH_4(w_{\iota 1})\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}F_j).
\end{aligned}$$

Therefore, we get that:

$$\begin{aligned}
H_4(w_\iota)A_{ID_u^*\|\iota}\theta_\iota^* &= H_4(w_\iota)A_{ID_u^*\|\iota}(\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}\theta_{\iota,j} + \nu_{\iota,t}\theta_{\iota,t}^*)\\
&= H_4(w_\iota)G\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}H_3(index\|j) + H_4(w_\iota)GA_c\\
&\quad \cdot\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}F_j + H_4(w_\iota)\nu_{\iota,t}A_{ID_u^*\|\iota}\theta_{\iota,t}^*.
\end{aligned}$$

Since $\mathcal{A}$'s forged response auditing proof $\mathsf{Proof}^* = \{\iota, \mu_\iota^*, \theta_\iota^*, w_i\}$ can satisfy the verification equation:

$$H_4(w_\iota)A_{ID_u^*\|\iota}\theta_\iota^* = G(H_4(w_\iota)\sum_{j\in\mathcal{J}}\nu_{\iota,j}\cdot H_3(index\|j) + A_c\mu_\iota^* - w_\iota).$$

According to the two forms of $H_4(w_\iota)A_{ID_u^*\|\iota}\theta_\iota^*$ above, we get that:

$$\begin{aligned}
H_4(w_\iota)G\nu_{\iota,t}H_3(index\|t) + GA_c\mu_\iota^* - Gw_\iota &= H_4(w_\iota)GA_c\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}F_j\\
&\quad + H_4(w_\iota)\nu_{\iota,t}A_{ID_u^*\|\iota}\theta_{\iota,t}^*.
\end{aligned}$$

As described before that $\mu_{\iota 1} = \beta_{\iota 1} + H_4(w_{\iota 1})\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}F_j$, where $\beta_{\iota 1} \leftarrow \mathsf{SamplePre}(A_c, T_c, w_{\iota 1}, \delta_0)$, thus:

$$\begin{aligned}
H_4(w_\iota)GA_c\sum_{j\in\mathcal{J},j\neq t}\nu_{\iota,j}F_j &= H_4(w_\iota)H_4(w_{\iota 1})^{-1}(GA_c\mu_{\iota 1} - GA_c\beta_{\iota 1})\\
&= H_4(w_\iota)H_4(w_{\iota 1})^{-1}(G(A_c\mu_{\iota 1} - w_{\iota 1})).
\end{aligned}$$

Set $\vartheta = H_4(w_\iota)H_4(w_{\iota 1})^{-1}(A_c\mu_{\iota 1} - w_{\iota 1}) \in \mathbb{Z}_q^n$. Finally, we get that:

$$H_4(w_\iota)G\nu_{\iota,t}H_3(index\|t) + GA_c\mu_\iota^* - Gw_\iota = G\vartheta + H_4(w_\iota)\nu_{\iota,t}A_{ID_u^*\|\iota}\theta_{\iota,t}^*.$$

Once receiving $chal = \{j, \nu_{i,j}\}_{j\in\mathcal{L}}$ from $\mathcal{B}$, $\mathcal{A}$ can tamper with F_t as F_t^*, and forge corresponding authenticator $\theta_{J,\,t}$ as $\theta_{\iota,t}^*$ with non-negligible probability ϵ, and $\mathcal{F}$ can further succeed in forging a different response auditing proof $\mathsf{Proof}^* = \{\iota, \mu_\iota^*, \theta_\iota^*, w_\iota\}$, naturally, here $\nu_{\iota,t} = 1$. Since $H_4(w_\iota) = 0$ or $H_4(w_{\iota 1}) = 0$ is negligible, thus the equation holds with non-negligible probability:

$$A_{ID_u^*\|\iota}\theta_{\iota,t}^* = G(H_3(index\|t) - (H_4(w_\iota))^{-1}(w_\iota - A_c\mu_\iota^* + \vartheta)).$$

As the right side of the equation above is actually an n-dimension vector in $\mathbb{Z}_q^n$, here we set $\hbar = G(H_3(index\|t) - (H_4(w_\iota))^{-1}(w_\iota - A_c\mu_\iota^* + \vartheta)) \in \mathbb{Z}_q^n$. In fact, to perform the attack process, $\hbar$ can be computed by $\mathcal{A}$ ahead of time. Thus, we get that $A_{ID_u^*\|\iota}\theta_{\iota,t}^* = \hbar$. Recall that $A = PR_\iota^*\cdots R_1^*$, suppose that the target identity ID_u^* satisfies $H_1(ID_u^*\|i) = R_i^*$, for $1 \le i \le i^* = \iota$.

By definition $A_{ID^*_u\|\imath} = A(R^*_1)^{-1}\cdots(R^*_\imath)^{-1} = P$, thus the case $P\theta^*_{\imath,t} = \hbar$ can occur with the probability $1/Q^\imath$. Moreover, the challenger $\mathcal{B}$ can successfully guess the time period when $\mathcal{B}$ breaks forward security with probability $1/\gamma$. Therefore, if $\mathcal{A}$ can generate the forged response auditing proof with non-negligible probability ϵ in some time period $\imath$, under the adaptively selective-ID attacks in the random oracle model, $\mathcal{B}$ has non-negligible probability $\epsilon' = \epsilon/(Q^\imath\gamma)$ to find the solution $\theta^*_{\imath,t}$ of the ISIS instance $(P, \hbar) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$, by running $\mathcal{A}$ (malicious cloud server) as a subroutine. □

Theorem 2. *The proposed scheme preserves privacy against the curious TPA, provided that the hardness assumption of SIS problem holds.*

Proof. As the combined message $\mu'_i = \sum_{j\in\mathcal{L}} \nu_{i,j} F_j$ is a linear combination of data blocks, once the combined message is returned to the TPA, it can solve the appropriate linear equations to derive the primitive data blocks, by using powerful computing devices. To handle this security issue, in **ProofGen**, the cloud server selects a random vector $w_i \leftarrow \mathbb{Z}_q^n$, runs $\mathsf{SamplePre}(A_c, T_c, w_i, \delta_0)$ to generate the signature β_i of w_i, termed as the random masking [29]. Thus the combined message μ'_i is blinded as $\mu_i = \beta_i + H_4(w_i)\mu'_i$. To further solve these linear equations, the TPA has to compute a valid signature β_i of w_i. However, according to the security proof in [15] based on the hardness assumption of SIS problem, without the trapdoor lattice basis T_c of the cloud server, the curious TPA can succeed in forging the valid signature β_i only with negligible probability. Therefore, given the response auditing proof $\mathsf{Proof} = \{i, \mu_i, \theta_i, w_i\}$, with the random masking technique, the proposed scheme preserves privacy against the curious TPA. □

4.3. Performance analysis

Now we give a detailed performance comparison in terms of the communication and verification overhead, among our scheme and existing key-exposure resilient auditing schemes [43,44]. All the experiments are run on a Linux server with Intel processor running at 2.70 GHz and 4 GB memory. All algorithms are written C with the MIRACL library version 5.6.1. The elliptic curve is the MNT curve, with the base field size of 159 bits and embedding degree 6. For the performance comparison, we set the security level for the challenge message to be 80 bits, such that $|\nu_{i,j}| = 80$ bits. We set $|p| = 160$ bits and the size of the an element in group G_1 to be 1024 bits. Meanwhile, to achieve the security of the hardness assumptions of ISIS, SIS problems, the parameters *m, n, q* need to satisfy $m \geq \lceil 2n\log q\rceil$. More specifically, we give an instance of the proposed scheme, the total number of time periods is γ, for simplicity, we set $t = \log\gamma + 2$. All the results of experiments are represented 30 trials on average.

We first give the numerical analysis of communication overhead comparison as follows. The communication overhead mainly consists of the challenge message overhead and the response auditing proof overhead. As described in [43], the communication overhead complexity of the response auditing proof is $O(\log\gamma)$, logarithmic in γ. While in our scheme and the scheme in [44], the communication overhead complexity of the response auditing proof is independent of γ. Moreover, the scheme in [44] has shown that its communication overhead is much smaller than that of [43]. Thus, we only focus on the communication overhead comparison between our scheme and the scheme in [44]. For consistency, we assume the number of data files is $\ell = 20$. In the scheme from [44], the communication overhead of challenge message is $c(|\ell| + |p|)$, while in our scheme, it is $c(|\ell| + 1)$, meaning the communication overhead of challenge message in [44] is nearly 9 times of ours. The communication overhead of the response auditing proof in scheme [44] is $2|G_1| + |p| = 2208$ bits, while in our scheme, as the response auditing proof is $Proof = \{\mu_i, \theta_i, w_i\}$, the communication overhead is $(2m + n)|q| = 4100$ bits, nearly double of [44]. We observe that our scheme enables the verification of the data file $\mathcal{F} = \{F_1, F_2, \cdots, F_\ell\} \in \mathbb{Z}_q^{m\times\ell}$, which is actually an $(m \times \ell)$-dimension matrix. While in [44], the data file $\mathcal{F} = \{F_1, F_2, \cdots, F_\ell\} \in \mathbb{Z}_p^\ell$ is actually an ℓ-dimension vector. It means that our scheme can deal with data files of size at least 16*m* times of that in [44], thus our communication overhead of response auditing proof is theoretically less than that of [44] amortized over data size. In addition, as described in [43], the private key size in each time period is logarithmic in γ, while in the scheme [44], the private key size is independent of γ. In our scheme, since we take advantage of $\mathsf{NewBasisdel}$ to delegate a short lattice basis, keeping the lattice dimension invariant. Thereby we can keep the private key size of each time period constant in **KeyUpdate**, which is also independent of γ. Therefore, compared with existing schemes, our scheme is much more efficient in the communication overhead.

Now, we compare the overhead of **ProofVerify**, and the hardness assumptions among our scheme and existing schemes. Firstly, we specify some notations to represent the computation of corresponding operations. T_{Pa} denotes a bilinear pairing operation $\hat{e}: G_1 \times G_1 \rightarrow G_T$, T_{Ex} denotes an exponentiation operation in G_1, T_{mu} denotes a common multiplication operation in G_1 or $\mathbb{Z}_q$, T_{ha} denotes a general hash function operation. The verification overhead comparison and the hardness assumptions are listed in Table 1, we can see that our scheme can achieve post-quantum secure, while existing schemes have no such security property. The implementation results in Fig. 2 demonstrate our scheme achieves better computational efficiency on the side of auditor, compared with existing schemes. This is mainly because our scheme is constructed with lattices, which only needs simple additions and multiplications over a moderate modulus, unlike time-consuming bilinear pairings and modular exponentiations in previous work.

5. Conclusion and future work

In this paper, we proposed a practical identity-based cloud storage public auditing scheme from lattices, achieving key exposure resistance. Particularly, we gave a formal security model of an identity-based public auditing scheme supporting

Table 1
Verification overhead comparison.

Schemes	ProofVerify	Hardness assumption
CSAKER [43]	$3T_{Pa} + (t+c+2)T_{Ex} +$ $(t+c+1)T_{mu} + (t+c+1)T_{ha}$	CDH
SKERCSA [44]	$3T_{Pa} + (c+2)T_{Ex} +$ $(c+2)T_{mu} + (c+1)T_{ha}$	CDH
Our scheme	$(n+c+1)T_{ha} + (cn+2mn+n^2+2n)T_{mu}$	ISIS

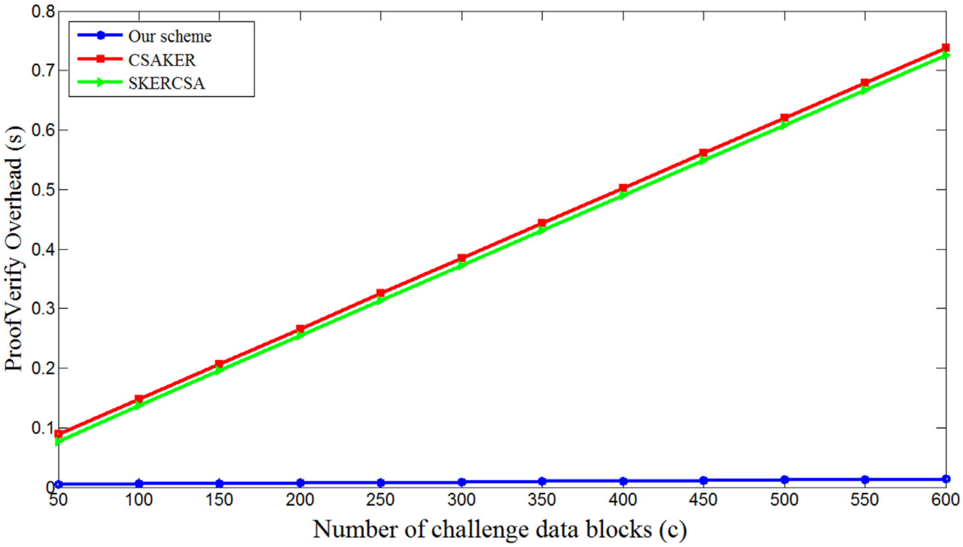

Fig. 2. ProofVerify overhead comparison.

forward security, and provided the detailed security proof for our scheme. The evaluation of performance comparison shows that our scheme is much more efficient on the side of the TPA indeed, and is more secure against adversaries with quantum computers. For future work, we will investigate how to construct an efficient lattice-based identity-based key-exposure resilient cloud storage public auditing schemes that not only protect against past keys in the event of key exposure, but also future keys.

Acknowledgement

This work is supported by National Key R&D Program of China (No.2017YFB0802000), National Natural Science Foundation of China (No.61872060), China Postdoctoral Science Foundation Funded Project (No.2017M623008), Sichuan Science and Technology Program (No. 2018GZ0102), Scientific Research Starting Project of SWPU (No.2017QHZ023), and the State Scholarship Fund of China Scholarship Council (CSC).

References

[1] M. Ajtai, Generating Hard Instances of the Short Basis Problem, in: Proceedings of Automata, languages and Programming, ICALP 1999, Springer Verlag Heidelberg, 1999, pp. 1–9.
[2] S. Agrawal, D. Boneh, X. Boyen, Lattice Basis Delegation in Fixed Dimension and Shorter-ciphertext Hierarchical IBE, in: Proceedings of Advances in cryptology-CRYPTO 2010, Springer-Verlag Heidelberg, 2010, pp. 98–115.
[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song, Provable Data Possession at Untrusted Stores, in: Proceedings of ACM Conference on Computer and Communications Security, ACM, 2007, pp. 598–609.
[4] J. Alwen, C. Peikert, Generating shorter bases for hard random lattices, Theory Comput. Syst. 48 (3) (2011) 535–553.
[5] D. Boneh, D.M. Freeman, Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-based Signature, in: Proceeding of Public Key Cryptography, PKC 2011, Springer-Verlag Heidelberg, 2011, pp. 1–16.
[6] A. Barsoum, M. Hasan, Provable multireplica dynamic data possession in cloud computing systems, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 485–497.
[7] D. Cash, D. Hofheinz, E. Kiltz, C. Peikert, Bonsai Trees, or How to Delegate a Lattice Basis, in: Proceedings of Advances in cryptology-EUROCRYPT 2010, Springer-Verlag Heidelberg, 2010, pp. 523–552.
[8] X. Chen, J. Li, X. Huang, J. Ma, W. Lou, New publicly verifiable databases with efficient updates, IEEE Trans. Dependable Secure Comput. 12 (5) (2015) 546–556.
[9] X. Chen, J. Li, J. Ma, J. Weng, W. Lou, Verifiable computation over large database with incremental updates, IEEE Trans. Comput. 65 (10) (2016) 3184–3195.
[10] W. Chen, H. Lei, K. Qi, Lattice-based linearly homomorphic signatures in the standard model, Theor. Comput. Sci. 634 (2016) 47–54.

[11] Y. Cui, Z. Lai, X. Wang, N. Dai, C. Miao, Quicksync: Improving Synchronization Efficiency for Mobile Cloud Storage Services, in: Proceedings of International Conference on Mobile Computing and Networks, ACM, 2015, pp. 592–603.
[12] Z. Cai, H. Yan, P. Li, Z. Huang, C. Gao, Towards secure and flexible EHR sharing in mobile health cloud under static assumptions, Cluster Comput. 20 (3) (2017) 2415–2422.
[13] X. Fu, X. Nie, T. Wu, F. Li, Large universe attribute based access control with efficient decryption in cloud storage system, J. Syst. Softw. 135 (2018) 157–164.
[14] C. Gao, S. Lv, Y. Wei, Z. Wang, Z. Liu, X. Cheng, M-SSE: An effective searchable symmetric encryption with enhanced security for mobile devices, IEEE Access. 2018, 10.1109/ACCESS.2018.2852329
[15] C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, in: Proceedings of the Fortieth Annual ACM symposium on Theory of Computing, 2008. 197–206.
[16] S. Gorbunov, V. Vaikuntanathan, D. Wichs, Leveled Fully Homomorphic Signatures from Standard Lattices, in: Proceedings of the forty-seventh annual ACM symposium on Theory of computing, 2015. 469–477.
[17] D. He, S. Zeadally, L. Wu, Certificateless public auditing scheme for cloud-assistedwireless body area networks, IEEE Syst. J. 99 (2017) 1–10.
[18] A. Juels, B.S. Kaliski, Pors: Proofs of Retrievability for Large Files, in: Proceedings of ACM Conference on Computer and Communications Security, ACM, 2007, pp. 584–597.
[19] K. Liang, C.K. Chu, X. Tan, D.S. Wong, C. Tang, J. Zhou, Chosen-ciphertext secure multi-hop identity-based conditional proxy re-encryption with constant-size ciphertexts, Theor. Comput. Sci. 539 (9) (2014) 87–105.
[20] T. Li, W. Chen, Y. Tang, H. Yan, A Homomorphic Network Coding Signature Scheme for Multiple Sources and Its Application in Iot, in: Security and Communication Networks, 2018, 10.1155/2018/9641273.
[21] J. Li, X. Huang, J. Li, X. Chen, Y. Xiang, Securely outsourcing attribute-based encryption with checkability, IEEE Trans. Parallel Distrib. Syst. 25 (8) (2014) 2201–2210.
[22] J. Li, J. Li, X. Chen, C. Jia, W. Lou, Identity-based encryption with outsourced revocation in cloud computing, IEEE Trans. Comput. 64 (2) (2015) 425–437.
[23] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, H. Wang, Signature Schemes with Efficient Protocols and Dynamic Group Signatures, in: Proceedings of Advances in cryptology-ASIACRYPT 2016, Springer Berlin Heidelberg, 2016. 373–403
[24] Microsoft.(2014). Key management. [Online]. Available: http://technet.microsoft.com/en-us/library/cc961626.aspx.
[25] D. Micciancio, O. Regev, Lattice-based Cryptography, in: Proceedings of Advances in cryptology-CRYPTO 2006, Springer Berlin Heidelberg, 2006. 131–141
[26] A. Shamir, Identity-based Cryptosystems and Signature Schemes, in: Proceedings of Advances in cryptology-CRYPTO 1984, Springer Berlin Heidelberg, 1984. 47–53
[27] P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26 (5) (1997) 1484–1509.
[28] H. Shacham, B. Waters, Compact Proofs of Retrievability, in: Proceedings of Advances in cryptology-ASIACRYPT 2008, Springer Berlin Heidelberg, 2008. 90–107
[29] C. Wang, S.M. Chow, Q. Wang, K. Ren, W. Lou, Privacy-preserving public auditing for secure cloud storage, IEEE Trans. Comput. 62 (2) (2013) 362–375.
[30] H. Wang, D. He, S. Tang, Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud, IEEE Trans. Inf. Forensics Secur. 11 (6) (2016) 1165–1176.
[31] F. Wang, Y. Hu, B. Wang, Lattice-based linearly homomorphic signature scheme over binary field, Science China Information Sciences 56 (11) (2013) 1–9.
[32] Z. Wu, L. Tian, P. Li, T. Wu, M. Jiang, C. Wu, Generating stable biometric keys for flexible cloud computing authentication using finger vein, Inf. Sci. (Ny) (2018). 431–447.
[33] B. Wang, B. Li, H. Li, Panda: public auditing for shared data with efficient user revocation in the cloud, IEEE Trans. Serv. Comput. 8 (1) (2015) 92–106.
[34] Y. Wang, Q. Wu, B. Qin, W. Shi, R.H. Deng, J. Hu, Identity-based data outsourcing with comprehensive auditing in clouds, IEEE Trans. Inf. Forensics Secur. 2 (12) (2017) 940–952.
[35] Q. Wang, C. Wang, K. Ren, W. Lou, J. Li, Enabling public auditability and data dynamics for storage security in cloud computing, IEEE Trans. Parallel Distrib. Syst. 22 (5) (2011) 847–859.
[36] H. Wang, Z. Zheng, L. Wu, P. Li, New directly revocable attribute-based encryption scheme and its application in cloud storage environment, Cluster Comput. 20 (3) (2017) 2385–2392.
[37] X. Wang, Y. Zhang, H. Zhu, L. Jiang, An Identity-based Signcryption on Lattice without Trapdoor, Journal of Universal Computer Science, 2018. To appear.
[38] Y. Yu, M.H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, G. Min, Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage, IEEE Trans. Inf. Forensics Secur. 12 (4) (2017) 767–778.
[39] L. Yang, Z. Han, Z. Huang, J. Ma, A remotely keyed file encryption scheme under mobile cloud computing, J. Netw. Comp. Appl. 106 (2018) 90–99.
[40] K. Yang, X. Jia, Data storage auditing service in cloud computing: challenges, methods and opportunities, World Wide Web 15 (4) (2012) 409–428.
[41] K. Yang, X. Jia, An efficient and secure dynamic auditing protocol for data storage in cloud computing, IEEE Trans. Parallel Distrib. Syst. 24 (9) (2013) 1717–1726.
[42] J. Yu, K. Ren, C. Wang, Enabling cloud storage auditing with verifiable outsourcing of key updates, IEEE Trans. Inf. Forensics Secur. 11 (6) (2016) 1362–1375.
[43] J. Yu, K. Ren, C. Wang, V. Varadharajan, Enabling cloud storage auditing with key-exposure resistance, IEEE Trans. Inf. Forensics Secur. 10 (6) (2015) 1167–1179.
[44] J. Yu, H. Wang, Strong key-exposure resilient auditing for secure cloud storage, IEEE Trans. Inf. Forensics Secur. 12 (8) (2017) 1931–1940.
[45] J. Yuan, S. Yu, Public integrity auditing for dynamic data sharing with multiuser modification, IEEE Trans. Inf. Forensics Secur. 10 (8) (2015) 1717–1726.
[46] Y. Zhang, X. Chen, J. Li, D.S. Wong, H. Li, I. You, Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing, Inf. Sci. (Ny) 379 (2017) 42–61.
[47] D. Zissis, D. Lekkas, Addressing cloud computing security issues, Future Generation Computer Systems 28 (3) (2012) 583–592.
[48] X. Zhang, C. Xu, C. Jin, R. Xie, Efficient forward secure identity-based shorter signature from lattice, Comput. Electr. Eng. 40 (6) (2014) 1963–1971.
[49] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, X. Zhang, Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation, IEEE Trans. Inf. Forensics Secur. 12 (3) (2017) 676–688.
[50] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, X. Lin, Healthdep: an efficient and secure deduplication scheme for cloud-assisted ehealth systems, IEEE Trans. Indust Inform. PP(99) (2018). 1–1.
[51] X. Zhang, C. Xu, Y. Zhang, Fuzzy identity-based signature scheme from lattice and its application in biometric authentication 11 (5) (2017) 2762–2777.
[52] X. Zhang, C. Xu, Y. Zhang, C. Jin, Efficient integrity verification scheme for medical data records in cloud-assisted wireless medical sensor networks, Wireless Personal Communications 96 (2) (2017) 1819–1833.
[53] X. Zhang, C. Xu, Y. Zhang, X. Zhang, J. Wei, Insecurity of a public proof of cloud storage from lattice assumption, Chinese J. Electron. 26 (1) (2017) 88–92.
[54] Y. Zhang, D. Zheng, R.H. Deng, Security and privacy in smart health: efficient policy-hiding attribute-based access control, IEEE Internet Things J. 5 (3) (2018) 2130–2145.