DOPIV: Post-Quantum Secure Identity-Based Data Outsourcing with Public Integrity Verification in Cloud Storage

Xiaojun Zhang, Jie Zhao, Chunxiang Xu, *Member, IEEE*, Huaxiong Wang, and Yuan Zhang, *Student Member, IEEE*

Abstract—Public verification enables cloud users to employ a third party auditor (TPA) to check the data integrity. However, recent breakthrough results on quantum computers indicate that applying quantum computers in clouds would be realized. A majority of existing public verification schemes are based on conventional hardness assumptions, which are vulnerable to adversaries equipped with quantum computers in the near future. Moreover, new security issues need to be solved when an original data owner is restricted or cannot access the remote cloud server flexibly. In this paper, we propose an efficient identity-based data outsourcing with public integrity verification scheme (DOPIV) in cloud storage. DOPIV is designed on lattice-based cryptography, which achieves post-quantum security. DOPIV enables an original data owner to delegate a proxy to generate the signatures of data and outsource them to the cloud server. Any TPA can perform data integrity verification efficiently on behalf of the original data owner, without retrieving the entire data set. Additionally, DOPIV possesses the advantages of being identity-based systems, avoiding complex certificate management procedures. We provide security proofs of DOPIV in the random oracle model, and conduct a comprehensive performance evaluation to show that DOPIV is more practical in post-quantum secure cloud storage systems.

Index Terms—Cloud storage, public verification, lattice-based cryptography, identity-based data outsourcing, post-quantum security

1 Introduction

Along with the rapid development of network and communication techniques, massive data are produced. These massive data need to be real-time processed with much more strong computation capability and greater storage space. With cloud storage services, cloud users can remotely outsource their massive data to the cloud server and access them via the Internet flexibly [1]. These services relieve cloud users from complicated local storage management and maintenance. Although cloud users enjoy great benefits from these services, some security concerns may impede cloud users to employ cloud storage [2], [3]. One of the most important security concerns is the data integrity [4]. Once these data are outsourced to the cloud server which is maintained by a cloud service provider (CSP), cloud users will lose physical control over their data. Thus, they will always worry about whether the outsourced data are kept intact, especially for those of important ones.

Actually, the cloud server is not fully trusted, and to keep its reputation intact, the cloud sever will not report any data loss incidents. To save the storage space, the cloud server may also delete a part of data that never be accessed by cloud users. In addition, due to some financial or political purposes, an external adversary may tamper with the outsourced data to achieve such goals and compromise its security. Consequently, it is necessary for cloud users to periodically check whether their outsourced data are stored properly. Since the quantity of the cloud storage data is huge, it is prohibitive for cloud users to download the entire data set to perform such integrity verification, and hence it is very impractical.

Recently, remote data public integrity verification becomes more and more significant due to the development of online storage systems [5], [6]. Public verification enables cloud users to delegate the data auditing tasks to a third-party auditor (TPA). The TPA periodically checks the remote data integrity without explicit knowledge of the entire data set, and informs cloud users that the data may

- *X. Zhang is with the School of Computer Science, Research Center for Cyber Security, Southwest Petroleum University, Chengdu, China, the Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China, the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 639798. E-mail: zhangxjdzkd2012@163.com.*
- *J. Zhao is with the School of Computer Science, Research Center for Cyber Security, Southwest Petroleum University, Chengdu, China. E-mail: zhaojswpu2017@163.com.*
- *C. Xu is with the Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China. E-mail: chxxu@uestc.edu.cn.*
- *H. Wang is with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 639798. E-mail: HXWang@ntu.edu.sg.*
- *Y. Zhang is with the Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China, and also with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: ZY_LoYe@126.com.*

Manuscript received 1 Oct. 2018; revised 28 July 2019; accepted 8 Sept. 2019. Date of publication 19 Sept. 2019; date of current version 4 Feb. 2022. (Corresponding author: Xiaojun Zhang.) Digital Object Identifier no. 10.1109/TSC.2019.2942297

be corrupted once the checking result fails [7]. In this case, cloud users will be free from the integrity verification burden. Also, the TPA needs to be equipped with strong computation capability, to execute time-consuming cryptographic operations, such as bilinear pairing and modular exponentiation operations. Additionally, some cloud users may store confidential or sensitive data (e.g., business contracts, medical records) in the cloud server. We should never ignore the fact that data privacy against the curious TPA in the integrity verification process is highly essential. With powerful computing devices, a semi-trusted TPA can succeed in recovering the primitive data based on the response auditing proof information from the cloud server [6].

While public verification has brought about significant benefits to cloud users, there are two hindrances in widely applying public verification into cloud storage. On one hand, with the rapid development of quantum computers, and the security of existing public verification schemes based on conventional cryptographic hardness assumptions will be threatened [8]. Some recent breakthrough research results [9], [10] indicate that quantum computers will come true in the near future. Thus, post-quantum secure public verification schemes will be more critical than ever before. On the other hand, most of public verification schemes are built upon the public key infrastructure (PKI). They are confronted with complex certificate management problem, since the certificate generation, storage, update and revocation procedures are very costly and cumbersome in practice [11]. On the contrast, the identity-based cryptographic systems [12] can reduce the complexity and avoid the establishment of the PKI. In such a system, a trusted key generator center (KGC) can generate the private key according to any known information of an individual identity, e.g., name, email or telephone number.

In addition to the aforementioned hindrances, we note that most of existing public verification schemes lack a controlled way of delegated outsourcing. In some scenarios, an original data owner who is restricted to access the remote cloud server, would like to delegate the task of data outsourcing to a dedicated proxy. For instance, some practical problems that happen in business affairs. When a business data manager is suspected of being involved into the commercial fraud, he/she will be taken away by the police. During this period of investigation, he/she will be restricted to access the network in order to guard against collusion, but the legal business affairs will go on. Since a great many commercial data are being generated and needed to be processed just in time, to avoid facing such loss of economic interest, the business data manger has to delegate a trusted proxy (e.g., a business secretary) to process his/her critical commercial data.

We can also consider another instance that always happens in medical affairs. As a chronic disease patient, he/she needs long-term treatments, and continuous recording of health status data with wireless medical sensor devices producing massive medical data. To get much more effective and professional treatments, the patient needs to delegate a trusted proxy (a medical worker) to process these medical data, and outsource them to the cloud server associated with the Electronic Health System (EHS). Since the medical data are the basis of all clinical diagnoses, any modifications will lead to error diagnosis, including severe consequences, such as death [13]. Thus, the patient also needs to verifiably guarantee that whether the outsourced medical data have been kept unchanged in the remote cloud server.

To cope with the above issues for secure outsourced data in clouds, in this paper, we propose an efficient identity-based data outsourcing with public integrity verification scheme in cloud storage. Our scheme is designed on lattice-based cryptography, which has very strong security proofs based on worst-case hardness [14]. Specially, the contributions of this work are elaborated as follows.

- We propose an efficient identity-based data outsourcing with public integrity verification scheme in cloud storage, called DOPIV. DOPIV relies on a lattice-based linearly homomorphic proxy-oriented signature, which is based on the hardness assumption of inhomogeneous small integer solution (ISIS) problem [15], thereby achieving secure against the quantum attacks. Moreover, DOPIV possesses the advantages of being identity-based systems, avoiding complex certificate management in the PKI.
- DOPIV achieves proxy-oriented secure data outsourcing. In DOPIV, an original data owner authorizes the proxy to process the critical data by generating the signed warrant. Once validating the signed warrant, the proxy can further generate the signatures of these data and outsource them to the cloud server. As the warrant includes the relative rights and information of an original data owner and a proxy, any unauthorized entity cannot process these data on behalf of the original data owner. In addition, we prove that DOPIV can guarantee proxy-oriented security. This means that any outside adversary cannot impersonate the original data owner to authorize the proxy to process the data and outsource them to the remote cloud server.
- We prove that DOPIV achieves storage correctness guarantee, and hence it is computationally infeasible for malicious cloud servers to generate a forged auditing proof that can pass the verification phase. We construct a random masking by using a preimage sampleable function [15], and integrate it into DOPIV so that a curious TPA cannot derive the primitive data blocks of the original data owner. Furthermore, we conduct a comprehensive performance evaluation. Compared with existing schemes, DOPIV is much more practical for post-quantum secure cloud storage systems. More specifically, without needing much more time-consuming cryptographic operations, such as bilinear pairing and modular exponentiation operations, the TPA performs the auditing tasks, in terms of delegation verification and integrity verification, only with simple addition and multiplication operations over a moderate modules. Thus, DOPIV dramatically decreases the computational costs on the side of the TPA.

The remainder of this paper is organized as follows. In Section 2, we review the related work. In Section 3, we first introduce lattice-based cryptography, then we define the

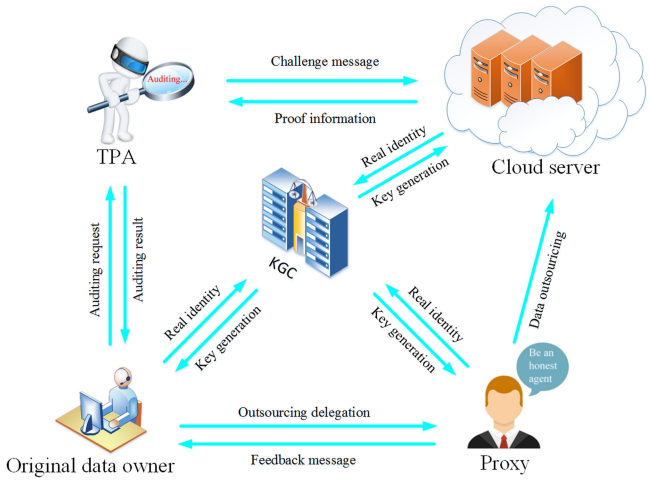


Fig. 1. The system model of DOPIV.

system model, formal definition, threat model, and design goals. In Section 4, we propose DOPIV. In Section 5, we prove the security of DOPIV. In Section 6, we conduct the performance evaluation. Finally, we draw the conclusions and present the future work in Section 7.

2 Related Work

With the rapid development of cloud computing technologies, cloud storage service becomes increasingly significant. Meanwhile, cloud users worry about the security and privacy of data stored in the cloud server [16], [17], [18] and [19], such as data confidentiality, and data reliability, data integrity. Recently, a majority of new encryption techniques [20], [21] and [22] have been proposed to ensure data confidentiality. Simultaneously, some other cryptographic techniques, e.g., [23], [24], [25], [26] and [27], have been employed to achieve flexible cloud storage data reliability. As to cloud users, one of the most important security issues is data integrity.

In order to ensure the integrity of outsourced data in the remote cloud server, Ateniese et al. [28] first proposed provable data possession (PDP) mechanism, which can resort to public auditors to perform such verification. Subsequently, Erway et al. [29] proposed a full dynamic PDP mechanism from authenticated flip table. Juels et al. [30] proposed proofs of retrievability (PoR) mechanism, thus the outsourced data stored in the remote cloud server can be retrieved and the data integrity can also be checked. However, the mechanism is actually a private verification. Shacham et al. [31] proposed a compact PoR mechanism with enhanced security, they gave a formal security definition of public verification. This model attracts many researchers and some PoR mechanisms have been designed in [32], [33] and [34].

Wang et al. [6] provided a privacy-preserving public auditing scheme which is based on BLS signature. This scheme employs a third-party auditor to check data integrity on behalf of cloud users, without downloading the entire data set. They integrate the random masking technique into the proposed scheme to preserve data privacy against curious auditors by solving appropriate linear equations. Following up Wang et al.'s work, many public verification schemes with new features have been proposed in [35], [36], [37], [38], [39], [40] and [41]. Specifically, the auditing schemes supporting data dynamic operations have been given in [35] and [37], whereas public integrity verification scheme in [38] employs the indistinguishability obfuscation technique to resist against malicious auditors. Cloud storage integrity verification schemes [39] and [40] with key-exposure resilience have been proposed to reduce the damage of cloud users' key exposure.

The aforementioned public verification schemes rely on public key infrastructure, and are confronted with considerable costs from complex certificate management problem. To avoid managing certificates, some public verification schemes [42], [43], [44], [45], [46] and [47] possessing the advantages of being identity-based cryptographic schemes have been proposed. Particularly, the cloud storage integrity verification schemes [42] and [43], supporting proxy-oriented data outsourcing, can flexibly enable original data owners to authorize the proxy to process the data and outsource them to the remote cloud server. An identity-based integrity auditing and data sharing with sensitive information hiding for cloud storage has been proposed in [46]. However, they need much more time-consuming cryptographic operations, such as bilinear pairing and modular exponentiation operations. Thus, the TPA needs to bear heavy computational costs in the auditing verification process.

Furthermore, according to the security analysis in [8], the conventional public-key cryptographic algorithms will be threatened, those public integrity verification schemes will be broken by quantum computers. Fortunately, lattice-based cryptography, as one of the most promising post-quantum cryptography, has very strong security proofs based on worst-case hardness [14] and [48]. Recently, lattice-based signature schemes with novel features have been proposed [49], [50], [51], [52] and [53]. Some of them [52], [53] with linearly homomorphic properties can be exploited to design public integrity verification schemes. In this paper, we explore how to combine a lattice-based linearly homomorphic proxy-oriented signature into an identity-based data outsourcing with public integrity verification scheme in cloud storage, which will achieve post-quantum security.

3 Preliminaries

3.1 System Model and Formal Definition

The system model of an identity-based data outsourcing with public integrity verification (DOPIV) is shown in Fig. 1. DOPIV consists of five different entities: original data owner, the proxy, cloud server, third-party auditor, and key generation center.

- *Original data owner*: His/her massive data need to be outsourced to the cloud server by the delegated proxy. The integrity of the outsourced data needs to be periodically checked by the TPA.
- *Cloud server*: It is managed by the cloud service provider. The cloud server has facilitative storage space and computation resource to maintain and manage these outsourced data.
- *The proxy*: It is authorized by an original data owner. Only when the proxy satisfies the warrant which is signed and issued by the original data owner, can

the proxy process these data and outsource them to the cloud server on behalf of the original data owner.
- *TPA*: It periodically checks the outsourced data integrity in the cloud server on behalf of the original data owner.
- *KGC*: Using the master public-secret key pair, the KGC generates the public-private key pair of any identity.

The formal definition of DOPIV consists of five phases, *Setup*, *KeyExtract*, *Proxy-oriented Signing KeyGen*, *Proxy-oriented TagGen*, and *Auditing Outsourced Data*.

Setup. The system initialization is a probabilistic polynomial-time algorithm (PPT). It takes the security parameter κ as input, and outputs the system public parameters and the master secret key Msk.

KeyExtract. This phase is performed by the KGC. Taking as input the master secret key Msk, the KGC generates the private key SK_{ID} corresponding to the identity ID.

Proxy-Oriented Signing KeyGen. This phase is an information interaction between an original data owner ID_o and a proxy ID_p. For the delegation of proxy-oriented signing rights, we denote a warrant w which records the delegation policy, valid period of delegation, the identities of an original data owner and the proxy.

Taking as inputs the system public parameters and the private key SK_{ID_o}, ID_o generates the signature of the warrant w, and sends it to ID_p. Once validating the signature of w, based on which, ID_p can further generate proxy-oriented signing private key SK_{pro} by using the private key SK_{ID_p}.

Proxy-Oriented TagGen. This phase is performed by the proxy ID_p. Taking as inputs the proxy-oriented signing private key SK_{pro}, big data file F, ID_p generates corresponding set of signatures Π. Finally, ID_p outsources them to the cloud server on behalf of ID_o, and deletes the local copy.

Auditing Outsourced Data. The data integrity verification phase consists of three polynomial-time algorithms, *Challenge*, *ProofGen*, and *ProofVerify*.

- *Challenge*: The PPT algorithm is performed by the TPA. It takes as inputs the system public parameters, outputs an auditing challenge message $chal$.
- *ProofGen*: The PPT algorithm is performed by the cloud server. It takes as inputs the challenge message $chal$, big data file F, and its corresponding set of signatures Π, outputs a response auditing proof information $Proof$ to TPA.
- *ProofVerify*: This deterministic polynomial-time algorithm is performed by the TPA. Taking as inputs the system public parameters, the response auditing proof information $Proof$, and the challenge message $chal$, the TPA outputs 1 if the integrity of the outsourced data can be verified as true; otherwise it outputs 0.

3.2 Threat Model and Design Goals

A practical identity-based data outsourcing with public integrity verification scheme (DOPIV) in cloud storage is confronted with three types of active attacks. First, a malicious cloud server may tamper with or delete the data and further generate the forged response auditing proof information, to pass the integrity verification process successfully. Second, an outside adversary may impersonate an original data owner to authorize a proxy to process the data, or abuse a delegation to process the data and outsource them to the cloud server. Third, a curious TPA may derive the primitive data of an original data owner by using powerful computing devices. Therefore, to enable DOPIV to be deployed in cloud storage systems efficiently under the threats of quantum-computing attacks, the following design goals should be achieved.

- *Storage correctness guarantee*: On behalf of an original data owner, the TPA can fulfil the integrity verification tasks without retrieving entire outsourced data set, so that a malicious cloud server cannot forge the response auditing proof information to convince the TPA that these data are kept intact indeed.
- *Proxy-oriented security*: Any outside adversary cannot impersonate an original data owner to authorize the proxy to process the data. Only if the proxy is authorized by the original data owner, and it satisfies the contents of the warrant, can the proxy process the data and outsource them to the cloud server on behalf of an original data owner.
- *Data privacy*: DOPIV should prevent a curious TPA from deriving the primitive data of the original data owner by solving proper linear equations [6].
- *High performance*: High performance is a practical requirement for deploying DOPIV in cloud storage systems. Especially on the side of the TPA, the computational costs and communication costs in integrity verification should be as little as possible. Moreover, enabling the TPA to check the data integrity without managing cloud users' certificates could be economic and favorable.
- *Post-quantum security*: Enabling DOPIV to resist against quantum computer attacks will has a great prospect in the near future.

As storage correctness guarantee is to ensure that the outsourced data are kept intact in the cloud server indeed, it is the most important security property in DOPIV. Compared with proxy-oriented security and data privacy against curious auditors, storage correctness guarantee owns much more complicated provable security proof process. Here we need to describe the security model of storage correctness ahead of time, and the security analysis of proxy-oriented security and data privacy against curious auditors will be provided later in Section 5.

Now, we provide the security model of storage correctness guarantee, we assume that an adversary $\mathcal{F}$ breaks storage correctness guarantee by interacting with the challenger $\mathcal{C}$ in the following phases.

Setup. $\mathcal{C}$ generates the system public parameters and returns them to $\mathcal{F}$.

$\mathcal{F}$ first announces to $\mathcal{C}$ the identity ID_p^* which will be challenged, it can adaptively perform the following queries.

KeyExtract query. $\mathcal{F}$ can adaptively query for the private key of any ID, and $\mathcal{C}$ generates the corresponding private key SK_{ID} and sends it to $\mathcal{F}$.

Proxy-Oriented Signing Private Key Query. Once $\mathcal{F}$ receives (w, θ_w) from an original data owner, where θ_w is the valid signature of the warrant w issued by the original data owner, $\mathcal{F}$ submits (w, θ_w) to $\mathcal{C}$ for querying the proxy-oriented signing

private key. $\mathcal{C}$ generates the corresponding proxy-oriented signing private key SK_{pro} and sends it to $\mathcal{F}$.

Proxy-Oriented TagGen Query. $\mathcal{F}$ can adaptively query the signatures of a series of data blocks $F_1, F_2, \ldots, F_\ell$ to $\mathcal{C}$. $\mathcal{C}$ generates the corresponding set of signatures for F_i (for $i = 1, \ldots, \ell$), and sends them back to $\mathcal{F}$.

Then $\mathcal{C}$ sends to $\mathcal{F}$ a challenge message $chal$, and requires $\mathcal{F}$ to provide a response auditing proof information under the challenge message $chal$.

Forgery Phase. After querying in a polynomial-time algorithm as before, the adversary $\mathcal{F}$ forges a response auditing proof information $Proof$ for the data blocks indicated by $chal$. If $\mathsf{ProofVerify}(ID_p^*, chal, Proof^*) = 1$, then $\mathcal{F}$ wins the above game.

We claim that DOPIV achieves storage correctness guarantee if the adversary $\mathcal{F}$ in above game makes the challenger $\mathcal{C}$ to accept its forged response auditing proof information only with negligible probability.

3.3 Lattice-based cryptography

Now, we introduce lattice-based cryptography, which has very strong security proofs based on worst-case hardness.

Let $B = \{b_1, \ldots, b_m\} \in \mathbb{R}^{m \times m}$ be a basis of the lattice Λ, which consists of m linearly independent vectors $b_1, \ldots, b_m$. The m-dimension full-rank lattice Λ generated by B is $\Lambda = \mathcal{L}(B) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{Z}^m, y = Bx \sum_{i=1}^{i=m} x_i b_i\}$. The Gram-Schmidt orthogonalization of the vectors $b_1, \ldots, b_m$ taken in that order denotes $\widetilde{B} = \{\widetilde{b_1}, \ldots, \widetilde{b_m}\}$, and $\|\widetilde{B}\|$ denotes the Gram-Schmidt norm of B.

Definition 1. *Given a prime q, a matrix $A \in \mathbb{Z}_q^{n \times m}$, and a vector $y \in \mathbb{Z}_q^n$, the q-modular integer lattices [54] are defined as follows:*

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

Let L be a subset of $\mathbb{Z}^m$, for any positive parameter $\sigma > 0$ and a vector υ, a Gaussian function on $\mathbb{R}^m$ centered at υ is $\forall z \in \mathbb{Z}^m$, $\varphi_{\sigma,\upsilon}(z) = \exp(-\pi\|z - \upsilon\|)^2/\sigma^2$, and $\varphi_{\sigma,\upsilon}(L) = \sum_{z \in L} \varphi_{\sigma,\upsilon}(z)$. The discrete Gaussian distribution over L with parameters σ and υ is $\mathcal{D}_{L,\sigma,\upsilon}(z) = \varphi_{\sigma,\upsilon}/\varphi_{\sigma,\upsilon}(L)$, $\forall z \in L$.

Now we define lattice-based hardness assumptions as follows.

Definition 2. *The inhomogeneous small integer solution problem is described below: Given a prime q, a matrix $A \in \mathbb{Z}_q^{n \times m}$, a syndrome $y \in \mathbb{Z}_q^n$ and a positive real number ζ, the goal of ISIS problem is to find a nonzero integer vector $e \in \mathbb{Z}^m$ such that $Ae = y \bmod q$ and $\|e\| \le \zeta$.*

In a similar way, the goal of the small integer solution (SIS) problem is to solve a nonzero integer vector $e \in \mathbb{Z}^m$ such that $Ae = 0 \bmod q$ 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 problems SIS, ISIS are as hard as approximating the SIVP problem in the worst case to within the factor $\zeta \cdot \tilde{O}(\sqrt{n})$.

The preimage sampleable function and lattice basis delegation technique are introduced as follows.

Lemma 1. *There exists a probabilistic polynomial-time algorithm* (PPT) *called* $\mathsf{TrapGen}(q, n)$ *in [55] 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 $\Lambda_q^\perp(A)$, and the euclidean norm of all the rows is bounded by $O(n \log n)$.*

Lemma 2. *For any prime q, and $m \ge \lceil 2n \log q \rceil$, the* PPT *algorithm* $\mathsf{SamplePre}(A, T_A, y, \sigma)$ *in [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 parameter $\sigma \ge \|\widetilde{T_A}\| \cdot \omega(\sqrt{\log m})$, and a vector $y \in \mathbb{Z}_q^n$, and outputs a sample from a distribution that is statistically close to $\mathcal{D}_{\Lambda_q^y(A),\sigma}$, where $\mathcal{D}_{\Lambda_q^y(A),\sigma}$ is the discrete Gaussian distribution over $\Lambda_q^y(A)$ with parameter σ.*

Now, we introduce the lattice basis delegation algorithm called $\mathsf{NewBasisDel}$ in [56], which is a key technique to generate the proxy-oriented signing private key in our scheme. $\mathsf{NewBasisDel}$ refers to the distribution $\mathcal{D}_{m \times m}$ on matrices in $\mathbb{Z}_q^{m \times m}$, which denotes $(\mathcal{D}_{\mathbb{Z}^m, \delta_R})^m$ conditioned on the resulting matrix R being $\mathbb{Z}_q$-invertible, where $\delta_R = \sqrt{n \log q} \cdot \omega(\sqrt{\log m})$.

Lemma 3. $\mathsf{NewBasisDel}$ *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 basis T_A of $\Lambda_q^\perp(A)$ and a secure parameter $\delta \ge \|\widetilde{T_A}\| \cdot \delta_R \sqrt{m} \omega(\log^{3/2} m)$, and outputs a short lattice basis T_B of $\Lambda_q^\perp(Q)$, where $Q = AR^{-1}$.*

To prove that our scheme achieves storage correctness guarantee, we needs a PPT algorithm $\mathsf{SampleR}$ to sample matrices from a distribution, which is statistically close to $\mathcal{D}_{m \times m}$ over $\mathbb{Z}_q^{m \times m}$. In addition, all our security proofs make use of a PPT algorithm $\mathsf{SampleRwithBasis}$ to simulate a random short lattice basis in the following lemma. For the details of $\mathsf{SampleR}$ and $\mathsf{SampleRwithBasis}$, the reader may refer to [56].

Lemma 4. *Let $m \ge \lceil 2n \log q \rceil$, and a prime $q \ge 3$. For all but at most a q^{-n} fraction of rank n matrix A in $\mathbb{Z}_q^{n \times m}$, the* PPT *algorithm* $\mathsf{SampleRwithBasis}(\mathsf{A})$ *outputs a low-norm matrix R in $\mathbb{Z}^{m \times m}$ sampled from a distribution which is statistically close to $\mathcal{D}_{m \times m}$, and a random short basis T_Q of $\Lambda_q^\perp(Q)$, where $Q = AR^{-1}$, satisfying $\|\widetilde{T_B}\| \le \delta_R/\omega(\sqrt{\log m})$ with overwhelming probability.*

4 The Proposed DOPIV

4.1 Overview of DOPIV

Now, we provide an overview of our identity-based data outsourcing with public integrity verification scheme from lattice assumptions in cloud storage. DOPIV consists of *Setup*, *KeyExtract*, *Proxy-oriented Signing KeyGen*, *Proxy-oriented TagGen*, and *Auditing Outsourced Data*. To construct the secure DOPIV scheme from lattice assumptions, we need to set secure parameters δ, σ for q-modular lattices in *Setup*. Moreover, we employ the trapdoor generation algorithm $\mathsf{TrapGen}$ to generate the master public-secret key pair of the KGC. In the phase of *KeyExtract*, the KGC can generate corresponding public-private key pair of any known information of an identity by using lattice basis delegation technique $\mathsf{NewBasisDel}$, which makes sure that the public key and private key sizes are invariant [56].

In the phase of *Proxy-oriented Signing KeyGen*, we first denote a warrant $w \in \{0,1\}^{\kappa_2}$, which contains concrete descriptive information of the delegation policy, validity period of delegation, and the corresponding identities of an original data owner and the proxy. An original data owner exploits the preimage sampleable function $\mathsf{SamplePre}(Q_{ID_o}, T_{ID_o}, u_w, \sigma)$ to generate a preimage $\theta_w \in \mathbb{Z}_q^m$. Here u_w is actually a hash function value of w, T_{ID_o} is actually a trapdoor of $\mathsf{SamplePre}$, the preimage θ_w can be considered as a signature of warrant w. Thus, this will guarantee that any other party cannot impersonate the original data owner to authorize the proxy to process the data. As the warrant w contains the real proxy's identity ID_p, thus, based on the signature θ_w of the warrant w, the proxy ID_p can further exploit $\mathsf{NewBasisDel}$ $(Q_{ID_p}, R_w, T_{ID_p}, \delta)$ to generate the proxy-oriented signing private key T_{pro}, which also makes sure that the proxy-oriented signing private key size is invariant. Since T_{ID_p} is actually a trapdoor of $\mathsf{NewBasisDel}$, any other party without being authorized by the original data owner in the warrant w cannot generate T_{pro}. T_{pro} is a critical factor to realize the function of outsourcing delegation. With the trapdoor T_{pro}, ID_p can generate signatures of the primitive data and upload them to the remote cloud server, on behalf of the original data owner.

In the phase of *Proxy-oriented TagGen*, to construct a proxy-oriented linearly homomorphic signature from lattice assumptions, the proxy ID_p first constructs a lattice-based additive hash function, which is actually an inner product computation of η_i and λ_j, where η_i is a linear transformation of each subfile F_i, λ_j is a hash function value related with the identity ID_p. Then, ID_p runs $\mathsf{SamplePre}(Q_{pro}, T_{pro}, \rho_i, \sigma)$ to generate a linearly homomorphic signature $e_i \in \mathbb{Z}_q^m$, under the proxy-oriented signing private key T_{pro}. Thus, all the corresponding linearly homomorphic signatures can be aggregated by the cloud server. Finally, all the corresponding signatures of subfiles as well as the signed warrant, can be outsourced to the remote cloud server by ID_p. Once validating the signature of warrant, the cloud server accepts and stores them.

In the phase of *Auditing Outsourced Data*, as similar to the model of [6], this phase also consists of *Challenge*, *ProofGen*, and *ProofVerify* algorithms. In particular, in *Challenge*, the challenge message is $chal = \{i, \beta_i\}_{i\in\Omega}$, where $\beta = (\beta_{l_1}, \ldots, \beta_{l_c}) \in \{0,1\}^c$, which dramatically decreases communication overhead of challenge message. In *ProofGen*, to further preserve data privacy against curious auditors, the cloud server employs $\mathsf{SamplePre}(Q_{ID_c}, T_{ID_c}, \xi, \sigma)$ to generate a signature h of a random vector $\xi \in \mathbb{Z}_q^n$ under its private key T_{ID_c}, here h and ξ can be considered as the random masking technique to blind the challenged combined message. Thus, any curious auditor cannot recover the primitive data of the original data owner. In *ProofVerify*, to check the data integrity, the TPA only needs to compute appropriate linear equations over a moderate modulus, thereby dramatically alleviating the auditing tasks.

4.2 The Construction of DOPIV

Now we propose the identity-based data outsourcing with public integrity verification scheme from lattice assumptions in cloud storage as follows.

Setup. Taking as input a security parameter κ, the system initialization determines the system parameters as follows.

1) Determine the discrete Gaussian distribution χ and secure Gaussian parameters δ, σ.
2) Choose six secure cryptographic hash functions: $H_1 : \{0,1\}^{\kappa_1} \to \mathbb{Z}_q^{m\times m}$, $H_2 : \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2} \times \mathbb{Z}_q^n \to \mathbb{Z}_q^n$, $H_3 : \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_1} \times \{0,1\}^{\kappa_2} \times \mathbb{Z}_q^m \to \mathbb{Z}_q^{m\times m}$, $H_4 : \{0,1\}^* \times \{0,1\}^* \to \mathbb{Z}_q^n$, $H_5 : \{0,1\}^{\kappa_1} \times \{0,1\}^* \to \mathbb{Z}_q^n$, and $H_6 : \{0,1\}^{\kappa_2} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^m \times \mathbb{Z}_q^n \to \mathbb{Z}_q$, where the outputs of H_1, H_3 are distributed in $\mathcal{D}_{m\times m}$.
3) Run $\mathsf{TrapGen}(q, n)$ to generate the KGC's master public key $A \in \mathbb{Z}_q^{n\times m}$ together with master secret key $T_A \in \mathbb{Z}_q^{m\times m}$ of $\Lambda_q^{\perp}(A)$.

The system public parameters are $\Sigma = (A, H_1, H_2, H_3, H_4, H_5, H_6, \delta, \sigma)$, the KGC secretly keeps the master secret key $\mathsf{Msk} = T_A$.

KeyExtract. Taking as inputs the master secret key T_A, the system public parameters Σ, and an original data owner's identity $ID_o \in \{0,1\}^{\kappa_1}$, the KGC generates the original data owner's private key as follows.

1) Compute $R_{ID_o} = H_1(ID_o)$, and compute $Q_{ID_o} = A(R_{ID_o})^{-1} \in \mathbb{Z}_q^{n\times m}$ as the public key of ID_o.
2) Run $\mathsf{NewBasisDel}(A, R_{ID_o}, T_A, \delta)$ to generate a random short lattice basis $T_{ID_o} \in \mathbb{Z}_q^{m\times m}$ of $\Lambda_q^{\perp}(Q_{ID_o})$ as the corresponding private key T_{ID_o} of ID_o.

In a similar approach, the KGC can generate the public-private key pair (Q_{ID_p}, T_{ID_p}) of the proxy $ID_p \in \{0,1\}^{\kappa_1}$, the public-private key pair (Q_{ID_c}, T_{ID_c}) of the cloud server $ID_c \in \{0,1\}^{\kappa_1}$, respectively.

Proxy-Oriented Signing KeyGen. In order to generate the proxy-oriented signing private key, an original data owner ID_o interacts with the proxy ID_p as follows.

1) ID_o creates the warrant $w \in \{0,1\}^{\kappa_2}$ according to its requirements. There is an explicit description of the delegation policy, valid period of delegation, the identities of an original data owner and the proxy, such that a verifier can use it as a part of verification information.
2) ID_o randomly chooses a vector $v_w \leftarrow Z_q^n$, computes $u_w = H_2(ID_o \| ID_p \| w \| v_w)$, and runs $\mathsf{SamplePre}(Q_{ID_o}, T_{ID_o}, u_w, \sigma)$ to generate $\theta_w \in \mathbb{Z}_q^m$. Then ID_o sends (w, v_w, θ_w) directly to the proxy ID_p. Here, everybody can get the signature of the warrant w, and verify its validity.
3) Once receiving (w, v_w, θ_w) from ID_o, ID_p validates the signed warrant w by computing $Q_{ID_o}\theta_w = u_w$, where $u_w = H_2(ID_o \| ID_p \| w \| v_w)$, θ_w is distributed in the distribution $\mathcal{D}_{\Lambda_q^{u_w}(Q_{ID_o}),\sigma}$. If the verification equation does not hold, ID_p rejects it and informs ID_o. Otherwise, ID_p goes on with computing $R_w = H_3(ID_o \| ID_p \| w \| \theta_w)$, and runs the algorithm $\mathsf{NewBasisDel}(Q_{ID_p}, R_w, T_{ID_p}, \delta)$ to generate the proxy-oriented signing private key T_{pro}, together with the corresponding proxy-oriented signing public key $Q_{pro} = Q_{ID_p}(R_w)^{-1}$ of ID_p.

Proxy-Oriented TagGen. Once ID_p satisfies the relative rights and information of the description in the warrant w, ID_p will process a big data file on behalf of ID_o. First, ID_p needs to preprocess the big data file F into ℓ subfiles $F = (F_1, F_2, \ldots, F_\ell)$, each subfile $F_i \in \mathbb{Z}_q^m$ has its corresponding name $N_i \in \{0,1\}^*$. With the proxy-oriented signing

private key T_{pro}, ID_p generates the signature of each subfile F_i as follows.

1) Compute $Q_{ID_c} = AR_{ID_c}^{-1} \in \mathbb{Z}_q^{n\times m}$, here ID_c is the identity of the cloud server, and compute $\eta_i = H_4(N_i\|i) + Q_{ID_c}F_i \in \mathbb{Z}_q^n$.
2) Compute the inner products $\rho_{i,j} = \langle \eta_i, \lambda_j \rangle$, $1 \le j \le n$, where $\lambda_j = H_5(ID_p\|j)$, set $\rho_i = (\rho_{i,1}, \ldots, \rho_{i,n})^\top \in \mathbb{Z}_q^n$.
3) Compute $Q_{pro} = Q_{ID_p}R_w^{-1} = AR_{ID_p}^{-1}R_w^{-1} \in \mathbb{Z}_q^{n\times m}$, run $\mathsf{SamplePre}(Q_{pro}, T_{pro}, \rho_i, \sigma)$ to generate $e_i \in \mathbb{Z}_q^m$.

Denote the set of signatures by $\Pi = \{(F_i, N_i, e_i), 1 \le i \le \ell\}$, ID_p outsources Π and (w, v_w, θ_w) to the cloud server. Once receiving (w, v_w, θ_w), the cloud server checks the validity by verifying whether the equation $Q_{ID_o}\theta_w = u_w$ holds, where $u_w = H_2(ID_o\|ID_p\|w\|v_w)$, θ_w is distributed in $\mathcal{D}_{\Lambda_q^{u_w}(Q_{ID_o}),\sigma}$. Then the cloud server checks whether ID_p satisfies w, which includes the relative rights and information of the description. If they hold, the cloud server accepts and stores (w, v_w, θ_w). Otherwise, the cloud server refuses to accept them.

Auditing Outsourced Data. The data integrity verification phase consists of three polynomial-time algorithms, *Challenge*, *ProofGen*, and *ProofVerify*.

- *Challenge*: To verify that the primitive big data file $F = (F_1, F_2, \ldots, F_\ell)$ keeps intact in the cloud server indeed, ID_o sends an auditing task request to the TPA. Then the TPA selects a random c-element subset $\Omega = \{l_1, \ldots, l_c\}$ of set $\{1, 2, \ldots, \ell\}$, and selects a random binary string $\beta = (\beta_{l_1}, \ldots, \beta_{l_c}) \in \{0,1\}^c$. The challenge message $chal = \{i, \beta_i\}_{i\in\Omega}$ locates the subfiles which need to be verified. Finally, the TPA sends $chal = \{i, \beta_i\}_{i\in\Omega}$ to the cloud server.
- *ProofGen*: Once receiving $chal = \{i, \beta_i\}_{i\in\Omega}$, the cloud server locates the corresponding outsourced subfiles of the primitive data file F, and computes $f' = \sum_{i=l_1}^{i=l_c} \beta_i F_i \in \mathbb{Z}_q^m$, $e = \sum_{i=l_1}^{i=l_c} \beta_i e_i \in \mathbb{Z}_q^m$. To further blind the aggregate data file f', the cloud server chooses a random vector $\xi \in \mathbb{Z}_q^n$, runs $\mathsf{SamplePre}(Q_{ID_c}, T_{ID_c}, \xi, \sigma)$ to generate the signature h of ξ, and computes $f = f' + hH_6(w\|v_w\|\theta_w\|\xi)$. Then, it sends $Proof = (f, e, \xi)$ as the response auditing proof information of storage correctness, and (w, v_w, θ_w) to the TPA.
- *ProofVerify*: Once receiving $Proof = (f, e, \xi)$ and (w, v_w, θ_w), the TPA first validates (w, v_w, θ_w) as before and checks whether ID_p satisfies w, which includes the relative rights and information of the description. Then the TPA performs as follows.
  1) For each $1 \le j \le n$, compute $\lambda_j = H_5(ID_p\|j)$, set $B = (\lambda_1, \ldots, \lambda_n)^\top$, and compute $\mu = B(\Sigma_{i=l_1}^{i=l_c}\beta_i H_4(N_i\|i) + Q_{ID_c}f - \xi H_6(w\|v_w\|\theta_w\|\xi)$.
  2) Check whether $Q_{pro}e = \mu \ mod \ q$ and $0 < \|e\| = \|\sum_{i=l_1}^{i=l_c} e_i\| \le c\sigma\sqrt{m}$ hold.

4.3 Correctness

The correctness of the verification equation is elaborated as follows.

$$
\begin{aligned}
&Q_{pro}e = Q_{pro}\textstyle\sum_{i=l_1}^{i=l_c}\beta_i e_i \\
&= \textstyle\sum_{i=l_1}^{i=l_c}\beta_i Q_{pro}e_i \\
&= \textstyle\sum_{i=l_1}^{i=l_c}\beta_i\rho_i \\
&= \textstyle\sum_{i=l_1}^{i=l_c}\beta_i(\langle\eta_i,\lambda_1\rangle,\ldots,\langle\eta_i,\lambda_n\rangle)^\top \\
&= \textstyle\sum_{i=l_1}^{i=l_c}\beta_i B\eta_i = B\sum_{i=l_1}^{i=l_c}\beta_i(H_4(N_i\|i) + Q_{ID_c}F_i) \\
&= B(\textstyle\sum_{i=l_1}^{i=l_c}\beta_i H_4(N_i\|i) + Q_{ID_c}(f - hH_6(w\|v_w\|\theta_w\|\xi))) \\
&= B(\textstyle\sum_{i=l_1}^{i=l_c}\beta_i H_4(N_i\|i) + Q_{ID_c}f - \xi H_6(w\|v_w\|\theta_w\|\xi)) \\
&= \mu \ mod \ q.
\end{aligned}
$$

Thus, the verification equation $Q_{pro}e = \mu \ mod \ q$ holds. For each $i \in \Omega = \{l_1, \ldots, l_c\}$, e_i is a signature of a subfile F_i, and thus $0 < \|e_i\| \le \sigma\sqrt{m}$ holds. Therefore, $0 < \|e\| = \|\sum_{i=l_1}^{i=l_c}\beta_i e_i\| \le c\sigma\sqrt{m}$ holds.

Actually, the cloud server and the TPA can always check the validity of the relative rights and information of the description in the warrant w, so that everybody can make sure whether the rights of the proxy are expired.

5 Security Proof of DOPIV

The security of DOPIV consists of the following three parts: storage correctness guarantee, proxy-oriented security, and data privacy against curious auditors.

Now, we first prove the storage correctness guarantee of DOPIV as follows.

Theorem 1. *DOPIV achieves storage correctness guarantee, provided that the hardness assumption of ISIS problem holds.*

Proof. We will demonstrate that if there exists any adversary $\mathcal{F}$ (a malicious cloud server) breaking storage correctness of DOPIV with a non-negligible probability ε, we can construct a challenger $\mathcal{C}$ to solve the hardness assumption of ISIS problem by running $\mathcal{F}$ as a subroutine, also with a non-negligible probability ε' in the random oracle model.□

Setup. First of all, $\mathcal{C}$ receives an ISIS instance $(U, \zeta) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$. $\mathcal{C}$ tries to solve the vector $e^* \in \mathbb{Z}_q^m$ such that $Ue^* = \zeta$. $\mathcal{C}$ sets six random oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$, $\mathcal{O}_{H_4}$, $\mathcal{O}_{H_5}$, and $\mathcal{O}_{H_6}$, respectively. Let V_{H_i} be the maximum number of $\mathcal{F}'s$ queries to $\mathcal{O}_{H_i}$ for $i = 1, 2, 3, 4, 5, 6$. $\mathcal{C}$ samples two random matrices $U_1^*, U_2^* \leftarrow \mathcal{D}_{m\times m}$ by running $\mathsf{SampleR}$, and randomly chooses $\upsilon_1^* \in [V_{H_1}]$, $\upsilon_3^* \in [V_{H_3}]$. Finally, $\mathcal{C}$ sets system public parameters $\Sigma = (A = UU_2^*U_1^*, \mathcal{O}_{H_1}, \mathcal{O}_{H_2}, \mathcal{O}_{H_3}, \mathcal{O}_{H_4}, \mathcal{O}_{H_5}, \mathcal{O}_{H_6})$, and sends Σ to $\mathcal{F}$.

$\mathcal{F}$ first announces to $\mathcal{C}$ the identity ID_p^* which will be challenged. We assume that $\mathcal{F}$ has queried the oracle $\mathcal{O}_{H_1}(ID)$ before ID is submitted to the KeyExtract query. Then $\mathcal{F}$ performs the following queries.

$\mathcal{O}_{H_1}$ query: $\mathcal{C}$ returns R_{ID}, if there exists $(ID, R_{ID}, Q_{ID}, T_{ID})$ in list L_1. If $\imath = \upsilon_1^*$, such that $ID = ID_p^*$, $\mathcal{C}$ sets $H_1(ID) = U_1^*$, adds $(ID_p^*, U_1^*, Q_{ID_p^*}, \perp)$ to L_1, and returns U_1^* to $\mathcal{F}$; otherwise, $\mathcal{C}$ runs $\mathsf{SampleRwithBasis}(A)$ to generate $R_{ID} \leftarrow \mathcal{D}_{m\times m}$ and a random short lattice basis T_{ID} for $Q_{ID} = A(R_{ID})^{-1}$, then $\mathcal{C}$ adds $(ID, R_{ID}, Q_{ID}, T_{ID})$ to L_1, and returns R_{ID} to $\mathcal{F}$.

$\mathcal{O}_{H_2}$ query: For the query on (ID_o, ID_p, w, v_w), $\mathcal{C}$ first checks if the value of H_2 was previously defined. If it was, the previously defined value is returned; otherwise, $\mathcal{C}$ randomly chooses a vector $u_w \leftarrow \mathbb{Z}_q^n$, adds it to list L_2, then returns it to $\mathcal{F}$.

$\mathcal{O}_{H_3}$ query: For the query on $(ID_o, ID_p, w, \theta_w)$, if $\imath = \upsilon_3^*$, such that $ID = ID_p^*$, $\mathcal{C}$ can search $(ID_p^*, U_1^*, Q_{ID_p^*}, \perp)$ in list L_1

to find $Q_{ID_p^*}$, $\mathcal{C}$ sets $H_3(ID_o\|ID_p^*\|w\|\theta_w) = U_2^*$, and adds $(ID_o, ID_p^*, w, \theta_w, U_2^*, Q_{pro^*}, \perp)$ to list L_3, where $Q_{pro^*} = Q_{ID_p^*}(U_2^*)^{-1}$, and returns U_2^* to $\mathcal{F}$; otherwise, $\mathcal{C}$ can search $(ID_p, R_{ID_p}, Q_{ID_p}, T_{ID_p})$ in list L_1 to find T_{ID_p}, randomly chooses $R_w \leftarrow \mathcal{D}_{m\times m}$, and proceeds to run NewBasisDel$(Q_{ID_p}, R_w, T_{ID_p}, \delta)$ to generate a random short lattice basis T_{pro} for $Q_{pro} = Q_{ID_p}(R_w)^{-1}$, then $\mathcal{C}$ adds $(ID_o, ID_p, w, \theta_w, R_w, Q_{pro}, T_{pro})$ to L_3, and returns R_w to $\mathcal{F}$.

The adversary $\mathcal{F}$ performs queries for $\mathcal{O}_{H_4}, \mathcal{O}_{H_5}, \mathcal{O}_{H_6}$ to $\mathcal{C}$. $\mathcal{C}$ can also answer these queries to $\mathcal{F}$ in a similar manner of the querying for $\mathcal{O}_{H_2}$, and $\mathcal{C}$ adds their hash function values to list L_4, L_5, L_6, respectively.

KeyExtract Query. For the query on ID, $\mathcal{C}$ first searches $(ID, R_{ID}, Q_{ID}, T_{ID})$ in list L_1. If it exists, $\mathcal{C}$ returns T_{ID} to $\mathcal{F}$; otherwise, $\mathcal{C}$ calls the $\mathcal{O}_{H_1}$ query, and returns the corresponding T_{ID} to $\mathcal{F}$.

Proxy-Oriented Signing Private Key Query. $\mathcal{F}$ submits $(ID_o, ID_p, w, \theta_w)$ to $\mathcal{C}$ for querying the proxy-oriented signing private key, with the restriction that $ID_p \neq ID_p^*$. $\mathcal{F}$ looks into list L_3, if it can find $(ID_o, ID_p, w, \theta_w, R_w, Q_{pro}, T_{pro})$ in L_3, $\mathcal{F}$ returns T_{pro} to $\mathcal{F}$; otherwise, $\mathcal{C}$ randomly chooses a matrix $R_w \leftarrow \mathcal{D}_{m\times m}$, runs NewBasisDel$(Q_{ID_p}, R_w, T_{ID_p}, \delta)$ to generate a random short lattice basis T_{pro} for $Q_{pro} = Q_{ID_p}(R_w)^{-1}$, then $\mathcal{C}$ adds $(ID_o, ID_p, w, \theta_w, R_w, Q_{pro}, T_{pro})$ to L_3, and returns T_{pro} to $\mathcal{F}$.

Proxy-Oriented TagGen Query. $\mathcal{F}$ submits a data file $F' = (F_1', F_2', \ldots, F_\ell')$, the proxy identity's ID_p, and cloud server's identity ID_c to the $\mathcal{C}$, where each subfile $F_i' \in \mathbb{Z}_q^m$ has its corresponding name $N_i' \in \{0,1\}^*$. $\mathcal{C}$ looks into list L_1 to get $(ID_c, R_{ID_c}, Q_{ID_c}, T_{ID_c})$, looks into list L_3 to get $(ID_o, ID_p, w, \theta_w, R_w, Q_{pro}, T_{pro})$, looks into L_4 to get $(N_i', i, H_4(N_i'\|i))$ for each $i = 1, \ldots, \ell$, and looks into L_5 to get $(ID_p, j, H_5(ID_p\|j))$ for each $j = 1, \ldots, n$. Then $\mathcal{C}$ performs as follows:

1) Compute each $\eta_i' = H_4(N_i'\|i) + Q_{ID_c}F_i' \in \mathbb{Z}_q^n$, compute $\rho_i' = (\rho_{i,1}', \ldots, \rho_{i,n}')^\top \in \mathbb{Z}_q^n$, where each inner product $\rho_{i,j}' = \langle \eta_i', \lambda_j \rangle, 1 \le j \le n, \lambda_j = H_5(ID_p\|j) \in \mathbb{Z}_q^n$.
2) Run SamplePre$(Q_{pro}, T_{pro}, \rho_i', \sigma)$ to generate $e_i' \in \mathbb{Z}_q^m$.

Finally, $\mathcal{C}$ returns $\Pi' = \{(F_i', N_i', e_i'), 1 \le i \le \ell\}$ to $\mathcal{F}$.

After querying for polynomial times as before, and when $\mathcal{C}$ generates $chal = \{i, \beta_i\}_{i\in\Omega}$ to the adversary $\mathcal{F}$, the adversary $\mathcal{F}$ performs as follows:

Forgery Phase. The adversary $\mathcal{F}$, as the role of the malicious cloud server, may try to perform tamper attacks in DOPIV. For simplicity, we assume that $\mathcal{F}$ can tamper with the data F_k as F_k^*, and forge corresponding signature e_k as e_k^* with a non-negligible probability. We consider the case that the malicious cloud server may delete some data for saving storage space, we term it as an instance of a tamper attack, where we can set $F_k^* = 0$, $e_k^* = 0$. Then $\mathcal{F}$ further tries to trick the TPA to believe that a forged response auditing proof can pass the verification process. More specifically, with the private key T_{ID_c}, $\mathcal{F}$ performs as follows.

1) Choose a random vector $\xi \leftarrow \mathbb{Z}_q^n$, and run SamplePre $(Q_{ID_c}, T_{ID_c}, \xi, \sigma)$ to generate the signature h of ξ. Compute the forged combined data $f^* = hH_6(M_w\| v_w\|\theta_w\|\xi) + \Sigma_{i\in\Omega, i\neq k}\beta_i F_i + \beta_k F_k^*$.
2) Compute the forged aggregate signature $e^* = \sum_{i\in\Omega, i\neq k}\beta_i e_i + \beta_k e_k^*$.

Finally, $\mathcal{F}$ sends the forged response auditing proof $Proof^* = (f^*, e^*, \xi)$ to $\mathcal{C}$. Suppose that $Proof^*$ is successfully forged, thus it can pass the correct verification equation $Q_{pro}e^* = B(\sum_{i\in\Omega}\beta_i H_4(N_i\|i) + Q_{ID_c}f^* - \xi H_6(M_\omega\|v_\omega\|\theta_\omega\|\xi))$, and $0 < \|e^*\| \le c\sigma\sqrt{m}$ holds. In fact, the equation $Q_{pro}e^* = Q_{pro}(\sum_{i\in\Omega, i\neq k}\beta_i e_i + \beta_k e_k^*)$. With regard to the combined message $\sum_{i\in\Omega, i\neq k}\beta_i F_i$ and the aggregate signature $\sum_{i\in\Omega, i\neq k}\beta_i e_i$, we consider that the cloud server has generated a valid response auditing proof information $Proof' = (f'', e', \xi')$ according to the challenge message $chal = \{i, \beta_i\}_{i\in\Omega, i\neq k}$ from $\mathcal{C}$ as before, where $\xi' \leftarrow \mathbb{Z}_q^n$ is another random vector, its valid signature is $h' \leftarrow$ SamplePre$(Q_{ID_c}, T_{ID_c}, \xi', \sigma)$, and we set $f'' = \sum_{i\in\Omega, i\neq k}\beta_i F_i + h'h_6'$, where $h_6' = H_6(w\|v_w\|\theta_w\|\xi')$, and $e' = \sum_{i\in\Omega, i\neq k}\beta_i e_i$. Thus the following equation holds:

$$\begin{aligned} Q_{pro}e' &= B\left(\sum_{i\in\Omega, i\neq k}\beta_i H_4(N_i\|i) + Q_{ID_c}f'' - \xi' h_6'\right) \\ &= B\left(\sum_{i\in\Omega, i\neq k}\beta_i H_4(N_i\|i) + Q_{ID_c}\left(\sum_{i\in\Omega, i\neq k}\beta_i F_i + h'h_6'\right) - \xi' h_6'\right). \end{aligned}$$

Therefore, we get that:

$$\begin{aligned} Q_{pro}e^* &= Q_{pro}\left(\sum_{i\in\Omega, i\neq k}\beta_i e_i + \beta_k e_k^*\right) \\ &= Q_{pro}\sum_{i\in\Omega, i\neq k}\beta_i e_i + Q_{pro}\beta_k e_k^* \\ &= B\left(\sum_{i\in\Omega, i\neq k}\beta_i H_4(N_i\|i) + Q_{ID_c}f'' - \xi' h_6'\right) + Q_{pro}\beta_k e_k^* \\ &= B\left(\sum_{i\in\Omega, i\neq k}\beta_i H_4(N_i\|i) + Q_{ID_c}\left(\sum_{i\in\Omega, i\neq k}\beta_i F_i \right.\right. \\ &\quad \left.\left. + h'h_6'\right) - \xi' h_6'\right) + Q_{pro}\beta_k e_k^* \\ &= B\left(\sum_{i\in\Omega, i\neq k}\beta_i H_4(N_i\|i) + Q_{ID_c}\sum_{i\in\Omega, i\neq k}\beta_i F_i\right) + Q_{pro}\beta_k e_k^* \end{aligned}$$

Since $\mathcal{A}'s$ forged response auditing proof information $Proof^* = (f^*, e^*, \xi)$ can pass the verification equation:

$$\begin{aligned} Q_{pro}e^* = B\Bigg(&\sum_{i\in\Omega}\beta_i H_4(N_i\|i) + Q_{ID_c}f^* \\ &- \xi H_6(M_\omega\|v_\omega\|\theta_\omega\|\xi)\Bigg). \end{aligned}$$

Here $h_6 = H_6(M_\omega\|v_\omega\|\theta_\omega\|\xi)$. According to the two forms of $Q_{pro}e^*$, we get that:

$$\begin{aligned} &B\beta_k H_4(N_k\|k) + BQ_{ID_c}f^* - B\xi h_6 = BQ_{ID_c} \\ &\cdot \sum_{i\in\Omega, i\neq k}\beta_i F_i + Q_{pro}\beta_k e_k^* \\ &= BQ_{ID_c}(f'' - h'h_6') + Q_{pro}\beta_k e_k^*. \end{aligned}$$

Set a vector $\hbar = Q_{ID_c}(f^* - f'' + h_1'h_6') \in \mathbb{Z}_q^n$. Thus, $B\beta_k H_4(N_k\|k) + B\hbar - B\xi h_6 = Q_{pro}\beta_k e_k^*$, $Q_{pro}\beta_k e_k^* = B(\beta_k H_4(N_k\|k) + \hbar - \xi h_6)$. Once receiving $chal = \{i, \beta_i\}_{i\in\Omega}$ from $\mathcal{C}$, $\mathcal{F}$ can tamper with the data F_k as F_k^*, forge the corresponding

TABLE 1
Communication Costs in Integrity Verification

Schemes	Communication overhead
ID-PUIC [42]	$3\|p\| + \|\ell\| + \|\mathbb{G}\|$
IBDO [43]	$(\ell + 2c)\|p\| + 2\|\mathbb{G}\|$
Our scheme	$(2m + n)\|q\| + (\|\ell\| + 1)c$

signature e_k as e_k^* with a non-negligible probability, and $\mathcal{F}$ can further succeed in forging a different response auditing proof $Proof^* = (f^*, e^*, \xi)$, naturally, where $\beta_k = 1$. Thus, the equation $Q_{pro}e_k^* = B(H_4(N_k\|k) + \hbar - \xi h_6)$ holds.

Since the right hand of the equation $Q_{pro}e_k^* = B(H_4(N_k\|k) + \hbar - \xi h_6)$ is actually an n-dimension vector in $\mathbb{Z}_q^n$, where we reset $\zeta = B(H_4(N_k\|k) + \hbar - \xi h_6) \in \mathbb{Z}_q^n$. Actually, to perform the attack process, ζ can be pre-computed by $\mathcal{F}$ ahead of time. Thus, we get that $Q_{pro}e_k^* = \zeta$. Recall that $A = UU_2^*U_1^*$, and the proxy-oriented signing public key $Q_{pro} = A(H_1(ID_p))^{-1}(H_3(ID_o\|ID_p\|w\|\theta_w))^{-1}$, thereby the equation $A(H_1(ID_p^*))^{-1}(H_3(ID_o\|ID_p^*\|w\|\theta_w))^{-1} = A(U_1^*)^{-1}(U_2^*)^{-1} = U$ holds with the probability $1/([V_{H_1}][V_{H_3}])$. Therefore, if $\mathcal{F}$ succeeds in forging a response auditing proof to pass the verification process with a non-negligible probability ε under the selective-ID security in the random oracle model, $\mathcal{C}$ will have a non-negligible probability $\varepsilon' = \varepsilon/([V_{H_1}][V_{H_3}])$ to find the solution e_k^* of the ISIS instance $(U, \zeta) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$ by running $\mathcal{F}$ (malicious cloud server) as a subroutine.

Then, we prove proxy-oriented security of DOPIV as follows.

Theorem 2. *DOPIV achieves proxy-oriented security, provided that the hardness assumption of ISIS problem holds.*

Proof. For simplicity, once an adversary captures the real authorized delegation information (w, v_w, θ_w) from an original data owner, we assume that the adversary has a non-negligible probability to forge a valid authorized delegation information (w', v'_w, θ'_w), without needing the private key T_{ID_o} of the original data owner. Thus, $Q_{ID_o}\theta'_w = H_2(ID_o\|ID_p\|w'\|v'_w) = h'_2$, where θ'_w is distributed in $\mathcal{D}_{\Lambda_q^{h'_2}(Q_{ID_o}),\sigma}$. In fact, the real authorized delegation information (w, v_w, θ_w) satisfies the equation $Q_{ID_o}\theta_w = H_2(ID_o\|ID_p\|w\|v_w) = h_2$, provided that θ_w is generated under the private key T_{ID_o} of the original data owner. Thus, if the adversary the adversary has a non-negligible probability to forge a valid authorized delegation information in the random model, it means that the adversary can succeed in finding the solution $\theta_w^* = \theta_w - \theta'_w$, such that $Q_{ID_o}\theta_w^* = h_2 - h'_2$, which actually reduces to solving the hardness assumption of ISIS problem. Therefore, DOPIV achieves proxy-oriented security, any outside adversary cannot generate the forged signature of the warrant, or cannot impersonate the original data owner to authorize the proxy to process the data, upload them to the cloud server. □

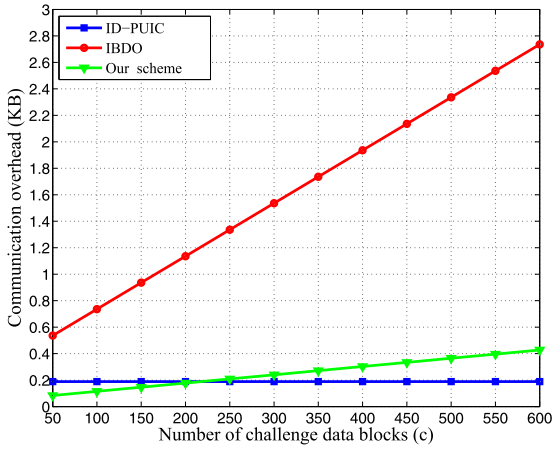


Fig. 2. The comparison of communication costs.

At last, we prove DOPIV achieves data privacy against curious auditors as follows.

Theorem 3. *DOPIV preserves data privacy against curious auditors, provided that the hardness assumption of SIS problem holds.*

Proof. Since the combined message $f' = \Sigma_{i=l_1}^{i=l_c}\beta_i F_i \in \mathbb{Z}_q^m$ is a linear combination of data blocks, a curious TPA can manage to recover these primitive data blocks by solving appropriate linear equations with powerful computing devices. To cope with this security issue, in *ProofGen*, the cloud server randomly selects a vector $\xi \leftarrow \mathbb{Z}_q^n$, runs SamplePre $(Q_{ID_c}, T_{ID_c}, \xi, \sigma)$ to generate the signature h of ξ, termed as a random masking technique [6]. Thus, the combined message f' is blinded as $f = f' + hH_6(w\|v_w\|\ \theta_w\|\xi)$, the response auditing proof is $Proof = (\xi, e, f)$. To further solve these linear equations, the curious TPA needs to compute a valid signature h of ξ. It means that, without the lattice basis T_{ID_c} of the cloud server, the curious TPA needs to succeed in forging the valid signature h in the random model, which actually reduces to solving the hardness assumption of SIS problem. Consequently, DOPIV achieves data privacy against curious auditors. □

6 Performance Evaluation

Now, we conduct a comprehensive performance evaluation compared with existing ID-PUIC scheme [42] and IBDO scheme [43], where ID-PUIC is an identity-based proxy-

TABLE 2
Computational Costs

Schemes	Delegation verification	Integrity verification
ID-PUIC [42]	$3Exp + 2mul + 2Hash$	$2Pair + (2c + 2)Exp + (c + 2)mul + 2Hash$
IBDO [43]	$2Pair + (\kappa_1 + \kappa_2 + 2)mul + (\kappa_1 + \kappa_2)Exp$	$4Pair + (c + 3)Hash + (\kappa_1 + c + m + 4)mul + (\kappa_2 + c + m + 2)Exp$
Our scheme	$Hash + nm \cdot mul$	$(n + c + 1)Hash + (n^2 + 2nm + n)mul$

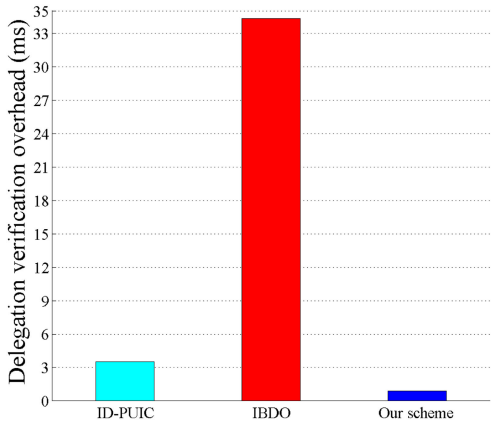


Fig. 3. The comparison of delegation verification.

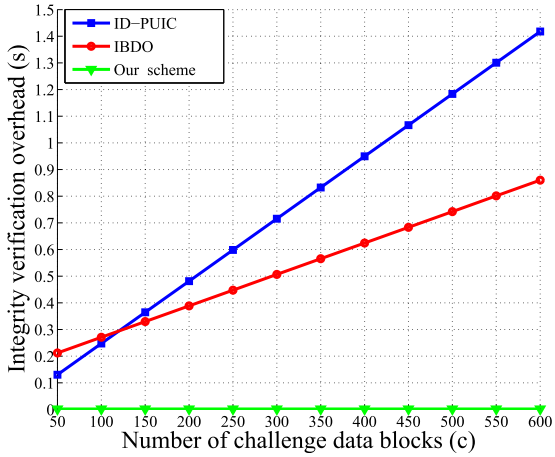


Fig. 4. The comparison of integrity verification.

oriented data uploading and remote data integrity checking scheme in public cloud, IBDO is an identity-based data outsourcing with comprehensive auditing scheme in clouds. Actually, they are designated on identity-based systems, which could avoid complex certificate management in the PKI. Simultaneously, ID-PUIC, IBDO could achieve proxy-oriented secure data outsourcing.

As our scheme also supports proxy-oriented data outsourcing, which resorts to the TPA to perform the data integrity verification. To deploy a practical DOPIV in mobile cloud storage systems, the key factors that affect the performance of a practical DOPIV in cloud storage systems are actually communication costs in integrity verification between the auditor and the cloud server, and the computational costs on the side of the auditor. Thus, in this paper, we focus on the performance evaluation of the auditing task, which consists of communication costs and computational costs in the integrity verification phase. Specifically, the computational costs include delegation verification overhead and integrity verification overhead. All the implementations are run on a Linux server with Intel processor running at 2.70 GHz and 4 GB memory. All algorithms are done in the C language and our code uses the MIRACL library version 5.6.1. The elliptic curve is MNT curve, its base field size is 159 bits and its embedding degree is 6. To achieve the security of the hardness of ISIS and SIS assumptions, the parameters m, n, q need to satisfy $m \geq \lceil 2n\log q \rceil$. More specifically, we give an instance of DOPIV, and compare it with existing schemes. All the results of experiments are representing 30 trials on average.

We first specify some notations to represent the corresponding cryptographic operations. Specifically, $|\mathbb{G}|$ denotes the bit length of an element in the cyclic group $\mathbb{G}$, $|p|$, $|q|$ denote the bit length of an element in $\mathbb{Z}_p$, $\mathbb{Z}_q$, respectively. In addition, $Pair$ denotes the running time of a bilinear pairing operation, Exp denotes the running time of a general module exponentiation, mul denotes the running time of a general multiplication, $Hash$ denotes the running time of a general hash function. We also denote $|\ell|$ by the number of the total data blocks, denote c by the number of the challenge data blocks, denote κ_1 by the size of an identity, and denote κ_2 by the size of a keyword. The communication costs in integrity verification phase are listed in Table 1. The implementation results in Fig. 2 demonstrate that DOPIV achieves a reasonable communication overhead in the integrity verification phase. Specifically, through detailed analysis, when the number of the challenge data blocks is less than 250, we can see that DOPIV is much more efficient than ID-PUIC and IBDO in communication overhead. With the growth of the number of the challenge data blocks, we get that the communication costs in IBDO is much higher than our DOPIV scheme and the ID-PUIC scheme. Although the ID-PUIC scheme has a little less communication costs than DOPIV scheme, our scheme achieves post-quantum security. Furthermore, the computational costs are listed in Table 2, including the delegation verification and integrity verification overhead. The implementation results in Fig. 3 demonstrate that DOPIV achieves much more efficient in delegation verification, thus anyone can check the validity of the signed warrant in the authorized delegation process with a less running time. Moreover, the implementation results in Fig. 4 show that the integrity verification overhead on the side of the auditor is much lightweight than the ones in existing schemes. This is mainly because DOPIV is based on lattice-based cryptography, without time-consuming bilinear pairing operations and modular exponentiations, the TPA only needs simple addition and multiplication operations over a moderate modulus. For instance, when the number of challenge data blocks is 350, the TPA actually only needs about 5.33 ms to fulfil such an auditing task.

Therefore, compared with existing proxy-oriented outsourcing with public verification schemes, DOPIV is much more practical in post-quantum secure cloud storage systems.

7 Conclusions and Future Work

In this paper, we have proposed a practical identity-based outsourcing with public integrity verification scheme (DOPIV) in cloud storage based on lattice-based cryptography, which achieves post-quantum security. DOPIV enables an original data owner to authorize a proxy to process the data and outsource them to the remote cloud server, and any TPA can check the data integrity on behalf of the original data owner. We provide the detailed security analysis of DOPIV, including storage correctness guarantee, proxy-oriented security, and data privacy against the curious TPA. The performance evaluation shows that DOPIV is more efficient on the side of the TPA indeed, and is more practical in the post-quantum secure cloud storage systems. In regards to future work, we will further investigate how to employ other lattice-based

cryptographic technologies to enhance public integrity verification for cloud storage systems in terms of security, performance, and functionality.

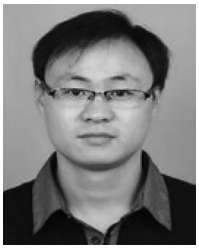
ACKNOWLEDGMENTS

This work is supported by National Key R&D Program of China (No.2017YFB0802000), National Natural Science Foundation of China (No.61872060, No.61902327), China Postdoctoral Science Foundation Funded Project (No.2017M623008), Singapore Ministry of Education under Research Grant MOE2016-T2-2-014(S) and by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative.

REFERENCES

[1] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud data protection for the masses," *Comput.*, vol. 45, no. 1, pp. 39–45, 2012.
[2] L. M. Kaufman, "Data security in the world of cloud computing," *IEEE Security Privacy*, vol. 7, no. 4, pp. 61–64, Jul./Aug. 2009.
[3] Y. Cui, Z. Lai, X. Wang, N. Dai, and C. Miao, "QuickSync: Improving synchronization efficiency for mobile cloud storage services," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 592–603.
[4] C. K. Chu, W. T. Zhu, J. Han, J. K. Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 50–57, Oct.–Dec. 2013.
[5] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Proc. Eur. Conf. Res. Comput. Security*, 2009, vol. 22, pp. 355–370.
[6] C. Wang, S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.
[7] M. Blum, W. Evans, P. Gemmell, and S. Kannan, "Checking the correctness of memories," *Algorithmica*, vol. 12, no. 2/3, pp. 225–244, 1994.
[8] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997.
[9] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O. Brien, "Quantum computers," *Nature*, vol. 464, no. 7285, pp. 45–53, 2012.
[10] Bloomberg, "IBM makes breakthrough in race to commercialize quantum computers," 2017. [Online]. Available: https://m.cacm.acm.org/
[11] J. K. Liu, H. A. Man, and W. Susilo, "Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model: Extended abstract," in *Proc. ACM Symp. Inf. Comput. Commun. Secur.*, 2007, pp. 273–283.
[12] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptographic Techn.*, 1984, pp. 47–53.
[13] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assistedwireless body area networks," *IEEE Syst. J.*, vol. 12, no. 1, pp. 64–73, Mar. 2018.
[14] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Proc. Annu. Int. Cryptology Conf.*, 2006, pp. 131–141.
[15] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, 2008, pp. 197–206.
[16] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.
[17] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "HealthDep: An efficient and secure deduplication scheme for cloud-assisted eHealth systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4101–4112, Sep. 2018.
[18] J. Xue, C. Xu, and L. Bai, "A distributed system for outsourced data storage and retrieval," *Future Generation Comput. Syst.*, vol. 99, pp. 106–114, 2019.
[19] X. Zhang, C. Xu, H. Wang, Y. Zhang, and S. Wang, "FS-PEKS: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted Industrial Internet of Things," *IEEE Trans. Dependable Secure Comput.*, doi: 10.1109/TDSC.2019.2914117.
[20] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2201–2210, Aug. 2014.
[21] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 425–437, Feb. 2015.
[22] X. Zhang, Y. Tang, H. Wang, C. Xu, Y. Miao, and H. Cheng, "Lattice-based proxy-oriented identity-based encryption with keyword search for cloud storage," *Inf. Sci.*, vol. 494, pp. 193–207, 2019.
[23] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 5, pp. 546–556, Sep./Oct. 2015.
[24] X. Chen, J. Li, J. Ma, J. Weng, and W. Lou, "Verifiable computation over large database with incremental updates," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 3184–3195, Oct. 2016.
[25] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Inf. Sci.*, vol. 379, pp. 42–61, 2017.
[26] X. Liu, B. Qin, R. H. Deng, and Y. Li, "An efficient privacy-preserving outsourced computation over public data," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 756–770, Sep./Oct. 2017.
[27] X. Fu, X. Nie, T. Wu, and F. Li, "Large universe attribute based access control with efficient decryption in cloud storage system," *J. Syst. Softw.*, vol. 135, pp. 157–164, 2018.
[28] 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. Security*, 2007, pp. 598–609.
[29] C. C. Erway, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Trans. Inf. Syst. Security*, vol. 17, no. 4, pp. 1–29, 2015.
[30] A. Juels and B. S. Kaliski, Jr, "Pors: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 583–597.
[31] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Security*, 2008, pp. 90–107.
[32] Y. Zhu, H. Wang, A. Hu, G. J. Ahn, H. Hu, and S. S. Yau, "Efficient provable data possession for hybrid clouds," in *Proc. 17th ACM Conf. Comput. Commun. Security*, 2010, pp. 756–758.
[33] Y. Zhu, H. Hu, G. J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2231–2244, Dec. 2012.
[34] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-replica provable data possession," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 2008, pp. 411–420.
[35] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1717–1726, Sep. 2013.
[36] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," *IEEE Trans. Services Comput.*, vol. 8, no. 1, pp. 92–106, Jan./Feb. 2015.
[37] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 8, pp. 1717–1726, Aug. 2015.
[38] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, and X. Zhang, "Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 676–688, Mar. 2017.
[39] 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.
[40] 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.
[41] Y. Zhang, C. Xu, X. Lin, and X. Shen, "Blockchain-based public integrity verification for cloud storage against procrastinating auditors," *IEEE Trans. Cloud Comput.*, doi: 10.1109/TCC.2019.2908400.
[42] H. Wang, D. He, and S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1165–1176, Jun. 2016.
[43] Y. Wang, Q. Wu, B. Qin, W. Shi, R. H. Deng, and J. Hu, "Identity-based data outsourcing with comprehensive auditing in clouds," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 12, pp. 940–952, Apr. 2017.

[44] H. Wang, "Identity-based distributed provable data possession in multicloud storage," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 328–340, Mar./Apr. 2015.
[45] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min, "Identity based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 767–778, Apr. 2017.
[46] W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 331–346, Feb. 2019, doi: 10.1109/TIFS.2018.2850312.
[47] X. Zhang, J. Zhao, C. Xu, H. Li, H. Wang, and Y. Zhang, "CIPPPA: Conditional identity privacy-preserving public auditing for cloud-based WBANs against malicious auditors", *IEEE Trans. Cloud Comput.*, doi: 10.1109/TCC.2019.2927219.
[48] Q. Lai, B. Yang, Y. Yu, Y. Chen, and J. Bai, "Novel smooth hash proof systems based on lattices," *Comput. J.*, vol. 61, no. 4, pp. 561–574, 2018.
[49] X. Boyen and Q. Li, "Towards tightly secure lattice short signature and ID-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Security*, 2016, pp. 404–434.
[50] L. Ducas and D. Micciancio, "Improved short lattice signatures in the standard model," in *Proc. Annu. Cryptology Conf.*, 2014, pp. 335–352.
[51] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang, "Signature schemes with efficient protocols and dynamic group signatures," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Security*, 2016, pp. 373–403.
[52] D. Boneh and D. M. Freeman, "Linearly homomorphic signatures over binary fields and new tools for lattice-based signature," in *Proc. Int. Workshop Public Key Cryptography*, 2011, pp. 1–16.
[53] W. Chen, H. Lei, and K. Qi, "Lattice-based linearly homomorphic signatures in the standard model," *Theoretical Comput. Sci.*, vol. 634, pp. 47–54, 2016.
[54] M. Ajtai, "Generating hard instances of the short basis problem," in *Proc. Int. Colloquium Autom. Lang. Program.*, 1999, pp. 1–9.
[55] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," *Theory Comput. Syst.*, vol. 48, no. 3, pp. 535–553, 2011.
[56] S. Agrawal, D. Boneh, and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Proc. Annu. Cryptology Conf.*, 2010, pp. 98–115.



Xiaojun Zhang received the BSc degree in mathematics and applied mathematics from Hebei Normal University, Shijiazhuang, China, in 2009, the MSc degree in pure mathematics from Guangxi University, Nanning, China, in 2012, and the PhD degree in information security from the University of Electronic Science Technology of China (UESTC), Chengdu, China, in 2015. He is a lecturer with the School of Computer Science, Southwest Petroleum University, Chengdu, China. He also works as a postdoctoral fellow in University of Electronic Science Technology of China from 2016. He is a research scholar with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. His research interests include cryptography, network security, and cloud computing security.

Jie Zhao received the BS degree in network engineering from Southwest Petroleum University, in 2017. He is currently working toward the postgraduate degree for MS degree in computer science and technology, in the School of Computer Science, Southwest Petroleum University. He is now presently engaging in cryptography, network security, cloud computing security and big data security.

Chunxiang Xu received the BSc, MSc and PhD degrees at Xidian University, in 1985, 1988 and 2004 respectively, P.R.China. She is presently engaged in information security, cloud computing security, and cryptography as a professor at University of Electronic Science Technology of China (UESTC). She is a member of the IEEE.

Huaxiong Wang received the PhD degree in mathematics from the University of Haifa, Israel, in 1996 and the PhD degree in computer science from the University of Wollongong, Australia, in 2001. He joined Nanyang Technological University, in 2006 and is currently an associate professor with the Division of Mathematical Sciences. He is also an honorary fellow at Macquarie University, Australia. His research interests include cryptography, information security, coding theory, combinatorics, and theoretical computer science. He has been on the editorial board of three international journals: the *Designs, Codes and Cryptography* (2006-2011), the *Journal of Communications* (JCM), and the *Journal of Communications and Networks*. He was the program cochair of Ninth Australasian Conference on Information Security and Privacy (ACISP 04), in 2004 and Fourth International Conference on Cryptology and Network Security (CANS 05), in 2005, and has served in the program committee for more than 70 international conferences. He received the inaugural Award of Best Research Contribution from the Computer Science Association of Australasia, in 2004.

Yuan Zhang received the BSc degree from the University of Electronic Science Technology of China (UESTC), in 2013, P.R.China. He is currently working toward the PhD degree in the School of Computer Science and Engineering, University of Electronic Science Technology of China. His research interests are cryptography, network security, and Cloud Computing security. He is a student member of the IEEE.