Lattice-based provable data possession in the standard model for cloud-based smart grid data management systems

Caixue Zhou, Lihua Wang and Lingmin Wang

Abstract
The smart grid is considered to be the next-generation electric power network. In a smart grid, there are massive data to be processed, so cloud computing is introduced into it to form a cloud-based smart grid data management system. However, with data no longer being stored locally, how to ensure the integrity of data stored in the cloud in the smart grid has become an urgent problem awaiting solution. Provable data possession has been proposed to solve this problem. With the development of quantum computer technology, quantum attacks-resistant cryptographic schemes are gradually entering people's horizons. Lattice cryptography can resist quantum attacks. In this article, a lattice-based provable data possession scheme is proposed for cloud-based smart grid data management systems. The scheme is proved unforgeable under the small integer solution hard assumption in the standard model. Compared with other two efficient lattice-based provable data possession schemes in the standard model, our scheme also shows efficiency.

Keywords
Cloud storage, lattice-based cryptography, provable data possession, smart grid, standard model

Date received: 7 March 2021; accepted: 3 February 2022

Handling Editor: Miguel Acevedo

Introduction
In recent years, the smart grid has gained more and more attention from many countries, and a great number of smart grid projects have been launched around the world. A smart grid is a new type of modern power grid integrating the advanced sensing measurement technology, information and communication technology, automatic control technology, analysis and decision technology, and energy power technology with grid infrastructure. As the next-generation electric power network, it is aimed to achieve reliability, security, economy, efficiency, environment friendliness as well as use safety.

A smart grid relies on the Internet of Things (IoT) technology. The core technologies of IoT cover the awareness of physical state, information representation, information transmission, and information processing, ranging from the sensor network to the upper application systems. In the aspects of communication, security, and the upper application of smart grid information system, IoT plays an important role. The sensor network technology can be used in the data and information acquisition of smart meters; the real-time and security communication technology can be used for the transmission of smart grid operation parameters, so that the operational data and power load data of the smart grid can be transmitted in real-time; the data storage and information representation technology can be used to store, manage, query, and organize huge
amounts of data of the smart grid; and the distributed data processing and task scheduling technology can be used for the security and stability analysis of the smart grid and the real-time deployment of energy after new energy integration. In a word, IoT has turned the power system from a relatively closed and self-contained control system into one that is part of a digital environment, which has not only improved the stability of the smart grid, but also made new energy sources such as wind power and nuclear power more easily integrated into the smart grid for unified planning and scheduling.

Nowadays, cloud computing provides users with massive computing and storage resources. Users can use it at a low cost without having to buy local equipment, saving a large amount of hardware, software, and maintenance costs. For cloud computing, the National Institute of Standards and Technology (NIST) defines three types of cloud services, that is, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). IaaS provides virtual machines or other infrastructure services such as storage resources; PaaS provides development platforms including software development kits, documentation, and testing environment; and SaaS provides application software based on cloud infrastructure, which users can use directly through browsers, and so on. At present, the research hotspots of cloud computing include job scheduling,3 cloud-based service broker routing policy,2 and so on.

In a smart grid, there are a large number of sensors deployed, exponentially increasing the collected data and thus greatly enlarging the data storage requirement. To deal with this issue, people combine the smart grid with cloud computing to form cloud-based smart grid data management systems.4

In terms of the security of cloud-based systems, Gou et al.5 analyzed various security issues and challenges in the cloud computing environment, which involve security policies, user-oriented security, data storage security, application security, and network security. In this article, we mainly focus on the data storage security.

In a cloud-based smart grid data management system, data are no longer stored locally. But if these data are changed intentionally or unintentionally without being detected in time by users, it will bring serious consequences to the operation of the smart grid. Therefore, ensuring the integrity of the data stored in the cloud in the smart grid has become an urgent problem awaiting solution. Ateniese et al.6 studied integrity checking for the first time. They put forward the concept of provable data possession (PDP) and proposed two concrete schemes. In their model, users can be ensured that their data are intact in the cloud without having to download the data from the cloud. It is a lightweight integrity probabilistic checking model. In their first scheme, if a file is divided into 10,000 blocks, of which 1% are tampered with, a user will be able to detect the change at a probability of 99% by just randomly selecting 460 blocks (4.6%) as the audit request. Following Ateniese et al.’s pioneering work, many other PDP schemes7,8 were proposed. To address the integrity of cloud-based smart grid data management systems, He et al.9 proposed a certificateless PDP scheme. However, Zhou9 later pointed out that He et al.’s scheme is insecure as a malicious cloud can cheat the users. In addition, He et al.’s scheme relies on the random oracle model10 and cannot withstand the quantum computer attacks.11

Our contributions

1. We propose a security model of the PDP scheme and summarize its security requirements.
2. We propose a quantum attack-resistant PDP scheme based on lattice for cloud-based smart grid data management systems and prove its unforgeable security in the standard model. We also show that the proposed scheme can satisfy other security requirements.
3. We compare our scheme with other two quantum attack-resistant PDP schemes in the standard model to show that our scheme is also efficient.

The rest of the article is organized as follows. In section “Related works,” we review some related work. In section “Preliminaries,” we describe some preliminaries about lattice. In section “Definition and security requirements of PDP,” we introduce the definition and security requirements of PDP. In section “The scheme,” we propose a concrete PDP scheme based on lattice. In section “Analyses of the scheme,” we analyze the security and efficiency of the proposed scheme. We conclude the article in section “Conclusion.”

Related works

In 2007, Ateniese et al.6 pioneered the idea of PDP and proposed two concrete schemes. However, their schemes only support private audit, that is, only cloud users can audit the data stored in the cloud, because in the audit process, some private information must be used. Later, Wang et al.12 put forward a public audit scheme. They removed the private information in the audit process, so that anyone can audit the data stored in the cloud. In this way, users can delegate their audit work to a third-party auditor (TPA) to reduce the burden of their own and the TPA can prove the data corruption to the court when a dispute occurs. Furthermore, Wang et al.13 pointed out that TPA can reveal users’ data by solving enough linear equations in
scheme [12] and proposed a privacy-preserving public PDP scheme by introducing random numbers.

All the above PDP schemes were proposed in the random oracle model, which, however, is just an ideal model. There is no random oracle in reality, and the random oracles are often replaced by some concrete hash functions in the real world. Some studies \(^{10}\) have shown that schemes secure in the random oracle model will not be still secure when the random oracles are replaced by concrete hash functions. To deal with this issue, Zhang et al.\(^ {14}\) proposed an efficient identity-based public PDP scheme in the standard model (without relying on the random oracle model).

With the rapid development of the quantum computer technology, all schemes based on factoring, discrete log, or bilinear pairings have become insecure under quantum computers, but schemes based on lattice can resist quantum computer attacks.\(^ {15}\) In 2016, NIST initiated a process to solicit, evaluate, and standardize post-quantum cryptographic algorithms around the world, marking the beginning of the post-quantum era. To deal with this issue, Xu et al.\(^ {16}\) proposed a lattice-based public PDP scheme. However, their scheme relies on the random oracle model. Chen et al.\(^ {17}\) proposed an efficient lattice-based PDP scheme in the standard model, which is based on the ideal lattice hard problem. Recently, Yang et al.\(^ {18}\) also proposed an efficient lattice-based PDP scheme in the standard model, which is based on the Ring Learning with Errors (RLWEs) hard problem. However, lattice with special structures such as ideal lattice and RLWE lattice is less secure than that without special structures.\(^ {19}\) Therefore, all the above schemes are not suitable for cloud-based smart grid data management systems.

In 2018, He et al.\(^ {4}\) proposed a certificateless PDP scheme for cloud-based smart grid data management systems. Thereafter, Zhou et al.\(^ {7}\) pointed out that He et al.’s scheme\(^ {4}\) is not secure as a malicious cloud can cheat users. Furthermore, He et al.’s scheme is not quantum attack-resistant and is reliant on the random oracle model.

### Preliminaries

In this section, we will give some definitions, theorems, and hard problem of lattice. The definitions will help the readers understand the concept of lattice better. The theorems show the existence of the algorithms TrapGen and SamplePre, which will be used in the design of our scheme. The hard problem will be used in the unforgeable proof of our scheme.

#### Notation

The lower-case letters denote the column vectors and the upper-case letters denote the matrices. \(A\) denotes the Gram–Schmidt orthogonalization of \(A\). \(|a|\) denotes the Euclidean norm of \(a\) and \(|A|\) denotes the largest Euclidean norm of the column vectors in \(A\).

#### Lattice

**Definition 1.** A lattice is a discrete additive subgroup of \(\mathbb{R}^n\).\(^ {20}\) It can be expressed as \(\Lambda = \Lambda(b_1, b_2, ..., b_n) = \left\{ \sum_{i=1}^{n} z_i b_i | z_i \in \mathbb{Z} \right\}\), where \(\{b_1, b_2, ..., b_n\} \in \mathbb{R}^m, m \gg n\) are \(n\) linearly independent vectors, which are called the basis of the lattice. The integers \(n\) and \(m\) are the rank and dimension of the lattice, respectively. If \(n = m\), the lattice is called a full rank lattice.

**Q-ary lattices**

**Definition 2.** Given a matrix \(A \in \mathbb{Z}^n \times m\) and a vector \(u \in \mathbb{Z}_q^n\), define \(\Lambda_q^+ (A) = \{ e \in \mathbb{Z}^m | Ae = 0 \mod q \}\) and \(\Lambda_q^-(A) = \{ e \in \mathbb{Z}^m | Ae = u \mod q \}\).\(^ {20}\)

**Discrete Gaussian distribution**

**Definition 3.** For a parameter \(\sigma > 0\) and a vector \(c \in \mathbb{R}^m\), the Gaussian function on \(\mathbb{R}^m\) centered at \(c\) is defined as \(\forall x \in \mathbb{R}^m, \rho_{\sigma, c}(x) = \exp(-\pi|x-c|^2/\sigma^2).\)^\(^ {20}\)

The discrete Gaussian distribution over \(\Lambda\) centered at \(c\) with standard deviation \(\sigma\) is defined as \(D_{\Lambda, \sigma, c}(x) = \rho_{\sigma, c}(x)/\sum_{v \in \Lambda} \rho_{\sigma, c}(v)\), where \(x \in \Lambda\).

**Trapdoor functions**

**Theorem 1.** Let \(n, m, q\) be the positive integers, \(m \gg n \log q\). There is a probabilistic polynomial time (PPT) algorithm TrapGen, which outputs the two matrices \((A, T) \in \mathbb{Z}_q^n \times m \times \mathbb{Z}_q^n \times m \leftarrow \text{TrapGen}(q, n, m)\), so that \(A\) is statistically close to a uniform matrix in \(\mathbb{Z}_q^n \times m\) and \(T\) is a basis for \(\Lambda^+_q(A)\) satisfying \(|T| \leq O(\sqrt{n \log q})\) and \(||T|| \leq O(n \log q)|.\)^\(^ {20}\)

**Theorem 2.** On input \(A \in \mathbb{Z}_q^n \times m\), a basis \(T\) of \(\Lambda^+_q(A)\), \(u \in \mathbb{Z}_q^n\) and \(\sigma \gg |T| ||\omega(\sqrt{\log m})||\), there is a PPT algorithm SampleP, which outputs a \(e \in \mathbb{Z}_q^n\) so that \(||e|| \leq \sigma \sqrt{m}\) and \(Ae = u \mod q\).\(^ {20}\)

**Hard lattice problem**

The security of our lattice-based PDP scheme relies on the hardness assumption of the small integer solution (SIS) problem.

**Definition 4.** SIS problem:\(^ {20}\) given \(n, m, q, \beta\), and a random matrix \(A \in \mathbb{Z}_q^n \times m\), one must find a non-zero vector \(v \in \mathbb{Z}_q^m\) so that \(Av \equiv 0 \mod q\) and \(|v|| \leq \beta\).
Definition and security requirements of PDP

In this section, we will give the algorithm constitution and design goals of PDP.

Definition of PDP

There are three participants in a PDP scheme, that is, Cloud server, Cloud user, and TPA, as illustrated in Figure 1. The process of PDP is as follows.

A user first divides his data to $n$ blocks and uses his private key to produce a tag for each block. Then, he uploads all the data blocks along with the tags to the cloud and deletes them locally. To ensure the data blocks are intact in the cloud, the user delegates TPA to audit the data blocks stored in the cloud periodically.

In the audit process, TPA randomly selects part of the data blocks for auditing. He sends the index numbers to audit the data blocks stored in the cloud periodically. If TPA can pass a verification equation in advance, then, TPA checks whether the proof can pass a verification equation in advance.

A PDP scheme consists of the following four algorithms:

1. Setup($1^k$): given a security parameter $1^k$, it generates a public/private key pair $(pk, sk)$.
2. TagSign$(sk, i, m_i)$: on input private key $sk$, index $i$ of file block $m_i$, and file block $m_i$, it generates a tag $t_i$ of $m_i$. It is run by the cloud user.
3. ProofGen$(pk, m_i, t_i, chal)$: on input public key $pk$, file blocks $m_i$’s, tags $t_i$’s, and a challenge $chal$, which contains a random index subset $I$ of total file blocks and random value set $W$, it generates a proof of data possession $V$. It is run by the cloud server.
4. VerifyProof$(pk, V, chal)$: on input public key $pk$, proof $V$, and challenge $chal$, it evaluates if $V$ is a correct proof of data possession for the blocks determined by $chal$. For public audit, it is run by TPA.

Design goals of PDP

1. Unforgeability: the PDP scheme must be existentially unforgeable against the adaptive chosen message attacks. The formal definition will be given in section “Analyses of the scheme.”
2. Public auditability: it does not need any private information when auditing the cloud. Therefore, users can delegate their audit tasks to TPA to release their own burden and prove the data corruption to the court when a dispute occurs.

The scheme

The scheme includes the following four algorithms. In the Setup algorithm, some common parameters are produced. In the TagSign algorithm, user produces a tag for each file block. Then, he transmits the file blocks along with the tags to the cloud and deletes them locally. In the ProofGen algorithm, TPA sends an audit request to the cloud and then the cloud responds a proof of data being intact to TPA. In the VerifyProof algorithm, TPA verifies the correctness of the proof. If it is correct, TPA can be sure that the data are intact.

Setup: the cloud user runs the TrapGen$(q, n, m)$ algorithm to generate a uniform matrix $A \in \mathbb{Z}_q^{n \times m}$ and a basis $T_A \in \mathbb{Z}_q^{m \times m}$ of $A_q \{n\}$ so that $|T_A| = O(\sqrt{n \log q})$. Similarly, the cloud server runs the TrapGen$(q, n, m)$ algorithm to generate a uniform matrix $Q \in \mathbb{Z}_q^{n \times m}$ and a basis $T_Q \in \mathbb{Z}_q^{m \times m}$ of $A_q \{Q\}$ so that $|T_Q| = O(\sqrt{n \log q})$. The cloud user chooses $2l + 2$ linearly independent vectors $c_0, c_1, \ldots, c_l, d_0, d_1, \ldots, d_l \in \mathbb{Z}_q^m$ randomly and produces a hash function $H : \mathbb{Z} \rightarrow \{0, 1\}^l$, where $l$ is the bit length of a file block. The public parameters are $pk = (A, Q, c_0, c_1, \ldots, c_l, d_0, d_1, \ldots, d_l)$. 

3. Privacy preserving: TPA should not deduce the users’ data by solving the linear equations when it audits the cloud.
TagSign: the message file is divided into $r$ blocks $\mu_i \in \{0, 1\}^\gamma$, $i = 1, 2, \ldots, r$, and $i \in Z$ is the index of $\mu_i$. For every block $i$, the cloud user computes $\eta_i = H(i) \in \{0, 1\}^\gamma$, $u_i = c_0 + \sum_{j=1}^{l-1} (\mu_j \cdot c_j) + d_0 + \sum_{j=1}^{l-1} (\eta_j \cdot d_j) \in Z^m$, and $u_i \equiv Q u_i \in Z_q^m$. Then, the cloud user computes $e_i \leftarrow \text{SamplePr} \langle A, T/A, u_i, \sigma \rangle \in Z^m$, where $\sigma = s + (1 + \sum_{j=1}^{l-1} (\mu_j \cdot c_j)) / (l + 1)$. The cloud user outputs $(\mu_i, e_i), i = 1, 2, \ldots, r$ to the cloud server. The cloud server can verify the correctness of these tags by $A e_i \equiv u_i \mod q$ and $||e_i|| \leq s / (1 + \sum_{j=1}^{l-1} (\mu_j \cdot c_j)) / (l + 1)$. 

ProofGen: to confirm the data in the cloud are intact, TPA selects a subset $J = \{i_1, i_2, \ldots, i_l\}$ of set $\{1, 2, \ldots, r\}$ and random elements $w_j \in Z_q^m, j = i_1, i_2, \ldots, i_l$. TPA sends $(j, w_j), j = i_1, i_2, \ldots, i_l$ to the cloud server. Then, the cloud server selects $v \in Z_q^m$ randomly. The cloud server computes $b \leftarrow \text{SamplePr} \langle \hat{Q}, T/A, v, \sigma \rangle \in Z^m$, $s = \sum_{k=1}^{l-1} w_k e_k \in Z_q^m$, and $t = b + \sum_{k=1}^{l-1} (w_k \sum_{j=1}^{l-1} (\mu_j \cdot c_j)) \in Z_q^m$. The cloud server sends $(s, t, v)$ to TPA. VerifyProof: TPA verifies whether the equation $As = \sum_{k=1}^{l-1} w_k Q(c_0 + d_0) + Q t - v - \sum_{k=1}^{l-1} (w_k Q \sum_{j=1}^{l-1} (\eta_j \cdot d_j))$ holds true or not.

**Analyses of the scheme**

**Security model of PDP**

In the following, we will give the security model of PDP, which will be used in the unforgeable security proof of our scheme.

The unforgeable security of a PDP scheme is defined through an interactive game between an adversary $A$ and a challenger $C$. The game consists of the following stages:

1. Setup: the challenger $C$ runs the setup algorithm to generate a public/private key pair $(pk, sk)$. $C$ gives $pk$ to $A$ and keeps $sk$ secret.
2. Query: $A$ can make $\text{TagSign}$ queries adaptively: $A$ selects a file block $m$ and sends it to $C$. $C$ runs the $\text{TagSign}$ algorithm to generate a tag of block $m$ and sends it back to $A$.
3. Challenge: $C$ generates a challenge $(I, W)$ and sends it to $A$, where $I$ is a subset of index numbers of file blocks and $W$ is a set of random numbers.
4. Forge: $A$ generates a proof $V$ for $(I, W)$ and returns $V$ to $C$.

If $V$ can pass the $\text{VerifyProof}$ algorithm and at least one query of $\text{TagSign}(sk, i, m_i)$ does not happen, where $i \in I$, $A$ wins the game.

**Definition 5.** A PDP scheme is secure if for any PPT adversary $A$, the probability that $A$ wins the above game is negligible.

**Security theorem**

According to the above security model of PDP, we prove that our scheme is unforgeable assuming that the SIS problem is hard.

**Theorem 3.** In the standard model, if there is an adversary $A$ that can break the unforgeability of the above scheme with probability $e$, then the SIS problem can be solved with $e = (1 − 2^{−\omega(\log^2 n)})$.

**Proof:** Given a random instance $\text{SIS}_{(m, m, q, 2\sigma)} = (A, n, m, s)$, where $A \in Z_q^m 
\times m$, the challenger $C$ is asked to find a short vector $c$ satisfying $Ac = 0 \mod q$ and $||c|| \leq 2\sqrt{m}, e \in Z^m$.

Setup: $C$ runs $(Q, T/O) \leftarrow \text{TrapGen}(q, n, m)$. $C$ chooses $e_i$ and $e_{ij}$ from $Z_q^m$ for $i = 0, 1, \ldots, l$, where $l$ is the bit length of a file block. Then, $C$ computes $c_i = Ac_i \mod q$ and $d_{ij} = Ae_{ij} \mod q$. $C$ computes vectors $c_0, c_1, \ldots, c_l \in Z_q^m$ so that $Q c_i = c_i \mod q$, $i = 0, 1, \ldots, l$ and vectors $d_0, d_1, \ldots, d_l \in Z_q^m$ so that $Q d_i = d_i \mod q$, $i = 0, 1, \ldots, l$. $C$ publishes $pk = (A, Q, c_0, c_1, \ldots, c_l, d_0, d_1, \ldots, d_l)$. $\text{TagSign}$ queries: $A$ produces a message block $\mu_i \in \{0, 1\}^\gamma$ and the index $i \in Z$ of $\mu_i$. $C$ computes $e_i = e_0 + \sum_{j=1}^{l-1} (\mu_i \cdot c_j) + e_0 + \sum_{j=1}^{l-1} (\eta_j \cdot e_j) \in Z^m$. $A$ gives $\bar{e}_i$ as the tag. From the adversary’s point of view, $e_i$ is a valid tag of $\mu_i$.

**Challenge:** $C$ selects a subset $J = \{i_1, i_2, \ldots, i_l\}$ of set $\{1, 2, \ldots, r\}$ and random elements $w_j \in Z_q^m, j = i_1, i_2, \ldots, i_l$. $C$ sends $(j, w_j), j = i_1, i_2, \ldots, i_l$ to $A$. $\text{Forge}$: $A$ generates a forged proof $(s, t, v)$ for $(j, w_j), j \in J$. According to Definition 5, at least one query of $\text{TagSign}(sk, i, m_i)$ must not happen, that is, $A$ generates a forged tag $e_i$ for file block $\mu_i$. If the forged tag is valid, then $Ac_i = u_i \mod q$ must hold. For the same file block $\mu_i$, $C$ computes $e_i = e_0 + \sum_{j=1}^{l-1} (\mu_i \cdot c_j) + e_0 + \sum_{j=1}^{l-1} (\eta_j \cdot e_j) \in Z^m$. Then, $A e_i \equiv u_i \mod q$. If $\bar{e}_i = e_i$, then $C$ aborts, or else we have $A(e_i - \bar{e}_i) \equiv 0 \mod q$ and $||e_i - \bar{e}_i|| \leq 2\sqrt{(1 + \sum_{j=1}^{l-1} (\mu_i \cdot c_j)) / (l + 1)} \sqrt{m} \leq 2\sqrt{m}$. That is, $C$ solves the SIS problem. The min-entropy of $\text{SamplePre}$ is $\omega(\log n)$. Therefore, the probability of $\bar{e}_i = e_i = 1 - 2^{−\omega(\log n)}$.

**Other design goals**

Some common design goals include public auditability and privacy preserving:
Our scheme is evident. In our \textbf{VerifyProof} algorithm, TPA does not need any private information in the verification equation, which means anyone can audit the data.

2. Privacy preserving: TPA should not deduce the users’ data by solving the linear equations when it audits the cloud. In our \textbf{ProofGen} algorithm, the cloud server sends proof \((s, t, v)\) to TPA. Note that \(t = b + \sum_{j \in I} (w_j \sum_{i \in J} (\mu_i [j] \cdot c_j)) \in \mathbb{Z}^m\) contains the original data blocks. If TPA selects the same index set of data blocks for audit \(c\) times, he can get \(c\) equations. However, \(b\) is unknown to him. The number of unknown variables is always greater than the number of equations. Therefore, he cannot reveal the user’s data blocks by solving the linear equations.

\textbf{Efficiency}

In this section, we will evaluate the efficiency of our scheme. Only efficient scheme has practical value. To the best of our knowledge, there have been only two lattice-based PDP schemes in the standard model till now, that is, schemes [17] and [18]. We compare the costs of computation, storage, and communication of our scheme with those of the other two. The computation cost of the \textbf{SamplePre} algorithm is equivalent to the cost of about \(m^2 + 2mn\) multiplications in \(\mathbb{Z}_q\) when the parity check matrix belongs to \(\mathbb{Z}_q \times m^{22}\). The comparisons are listed in Tables 1 and 2, where \(c\) denotes the number of audited blocks, and \(l\) denotes the bit length of a file block.

To provide more direct comparisons, we take \(n = 391, q = 2.34e + 10, m = 71,380\) for our scheme (non-ideal lattice) and \(n = 256, q = 7.21e + 16, m = 2204\) for schemes [17] and [18] (ideal lattice).\footnote{With these parameters adopted, its security level is about equivalent to that of 88-bit symmetric key cryptosystem. In addition, we take \(l = 160\). Therefore, we get Tables 3 and 4. From Table 3, we can get Figure 2 for cloud server’s running time and Figure 3 for TPA’s running time. From Table 3, we can see that scheme [17] is the most efficient one in the \textbf{TagSign} stage. From Table 4, we can see that scheme [17] is the most efficient one in the storage of \(Pk + Sk\) and that ours is the most efficient one in the storage of \textbf{Tags} and the communication of \textbf{Tags} and \textbf{Proof}. From Figure 2, we can see that ours is the most efficient one in cloud server’s running time (\textbf{ProofGen} stage). And according to Figure 3, scheme [17] is the most efficient one in TPA’s running time.}

1. Public auditability: the public auditability of our scheme is evident. In our \textbf{VerifyProof} algorithm, TPA does not need any private information in the verification equation, which means anyone can audit the data.

2. Privacy preserving: TPA should not deduce the users’ data by solving the linear equations when it audits the cloud. In our \textbf{ProofGen} algorithm, the cloud server sends proof \((s, t, v)\) to TPA. Note that \(t = b + \sum_{j \in I} (w_j \sum_{i \in J} (\mu_i [j] \cdot c_j)) \in \mathbb{Z}^m\) contains the original data blocks. If TPA selects the same index set of data blocks for audit \(c\) times, he can get \(c\) equations. However, \(b\) is unknown to him. The number of unknown variables is always greater than the number of equations. Therefore, he cannot reveal the user’s data blocks by solving the linear equations.

\begin{table}[h]
\centering
\caption{Computation costs.}
\begin{tabular}{|c|c|c|c|}
\hline
Schemes & \textbf{TagSign} & \textbf{ProofGen} & \textbf{VerifyProof} \\
\hline
[17] & \(mn^2\) & \(2cmn^2\) & \(mn^2\) \\
[18] & \(m^2 + 2mn + n^2m^2\) & \(2cm(n^2 + n)\) & \(n^2m + cmn\) \\
Our scheme & \(m^2 + 3mn\) & \(m^2 + 2(n + c)m\) & \((c + 1)(m + 1)n + mn\) \\
\hline
\end{tabular}
\end{table}

Note: \(n\) and \(m\) are of the same meaning as in the scheme, and \(c\) denotes the number of audited blocks.

\begin{table}[h]
\centering
\caption{Storage and communication costs.}
\begin{tabular}{|c|c|c|c|}
\hline
Schemes & \(Pk + Sk\) & \textbf{Tag-Length} & \textbf{Proof-Length} \\
\hline
[17] & \(mn \log q\) & \(nm \log q\) & \(2nm \log q\) \\
[18] & \(3nm \log q\) & \(nm \log q\) & \(2nm \log q\) \\
Our scheme & \(2(n + m + 1) \cdot m \log q\) & \(m \log q\) & \((2m + n) \log q\) \\
\hline
\end{tabular}
\end{table}

Note: \(n, m,\) and \(q\) are of the same meaning as in the scheme, and \(l\) denotes the bit length of a file block.

\begin{table}[h]
\centering
\caption{Computation costs (multiplications in \(\mathbb{Z}_q\)).}
\begin{tabular}{|c|c|c|c|}
\hline
Schemes & \textbf{TagSign} & \textbf{ProofGen} & \textbf{VerifyProof} \\
\hline
[17] & \(1.44e8\) & \(c \times 2.88e8\) & \(1.44e8\) \\
[18] & \(3.18e11\) & \(c \times 2.9e8\) & \(1.44e8 + 5.6e5 \times c\) \\
Our scheme & \(5.18e9\) & \(5.1e9 + (391 + c) \times 1.4e5\) & \((c + 2) \times 2.8e7\) \\
\hline
\end{tabular}
\end{table}

Note: \(c\) denotes the number of audited blocks, and \(e\) denotes 10. The number behind \(e\) denotes the exponent.
(VerifyProof stage). Therefore, it can be concluded that our scheme is efficient.

**Conclusion**

To resist quantum computer attacks, a lattice-based PDP scheme is proposed for cloud-based smart grid data management systems to ensure the integrity of cloud data. We prove that the scheme is secure under the SIS assumption in the standard model, and performance analysis shows that the scheme is efficient.

**Acknowledgements**

The authors thank Ms Yan Di for checking the article.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Scientific and Technological Research Project of Jiangxi Provincial Education Department of China under grant GJJ201807, the National Natural Science Foundation of China under grant 62041603, and the key Project of Jiangxi Provincial Natural Science Foundation of China under grant 2020ACBL202005.

**ORCID iD**

Caixue Zhou [https://orcid.org/0000-0002-7924-2584](https://orcid.org/0000-0002-7924-2584)

**References**

1. Zhang Y, Chen W and Gao WJ. A survey on the development status and challenges of smart grids in main driver countries. *Renew Sustain Energ Rev* 2017; 79: 137–147.
2. Jami SI and Munir S. Current trends in cloud computing for data science experiments. *Int J Cloud Appl Comput* 2021; 11(4): 80–99.
3. Manasrah AM, Aldomi A and Gupta BB. An optimized service broker routing policy based on differential

### Table 4. Storage and communication costs (bit).

| Schemes | $\text{Pk} + \text{Sk}$ | Tag-Length | Proof-Length |
|---------|-------------------------|------------|--------------|
| [17]    | 3.16e7                  | 3.16e7     | 6.32e7       |
| [18]    | 9.48e7                  | 3.16e6     | 6.32e7       |
| Our scheme | 1.77e11              | 2.46e6     | 4.93e6       |

Note: $e$ denotes 10. The number behind $e$ denotes the exponent.
4. He DB, Kumar N, Zeadally S, et al. Certificateless provable data possession scheme for cloud-based smart grid data management systems. *IEEE Trans Ind Inform* 2018; 14(3): 1232–1241.

5. Gou ZL, Yamaguchi S, and Gupta BB. Analysis of various security issues and challenges in the fog computing environment: a survey. In: Gupta BB, Agrawal DP, and Yamaguchi A (eds) *Handbook of research on modern cryptographic solutions for computer and cyber security*. Hershey, PA: IGI Global, 2016, pp.393–419.

6. Ateniese G, Burns R, Curtmola R, et al. Provably data possession at untrusted stores. In: *Proceedings of the 2007 ACM conference on computer and communications security (CCS)*, Alexandria, VA, 28–31 October 2007, pp.598–609. New York: ACM.

7. Kang BY, Wang JQ, and Shao DY. Certificateless public auditing with privacy preserving for cloud-assisted wireless body area networks. *Mob Inform Syst* 2017; 2017: 2925465.

8. Yu HY, Yang Z, and Sinnott RO. Decentralized big data auditing for smart city environments leveraging blockchain technology. *IEEE Access* 2018; 7: 6288–6296.

9. Zhou CX. Security analysis of a certificateless provable data possession scheme with privacy preserving for cloud-based smart grid data management system. *Int J Netw Secur* 2020; 22(4): 584–588.

10. Canetti R, Goldreich O, and Halevi S. The random oracle methodology, revisited. *J ACM* 2004; 51(4): 557–594.

11. Shor P. Polynomial-time algorithms for prime factorization and discrete logarithms in a quantum computer. *SIAM J Comput* 1997; 26(5): 1484–1509.

12. Wang Q, Wang C, Li J, et al. Enabling public verifiability and data dynamic for storage security in cloud computing. In: *Proceedings of the 14th European symposium on research in computer security (ESORICS)*, Saint-Malo, 21–23 September 2009, pp.355–370. New York: Springer.

13. Wang C, Wang Q, Ren K, et al. Privacy-preserving public auditing for data storage security in cloud computing. In: *Proceedings of the 29th IEEE international conference on computer communications (INFOCOM)*, San Diego, CA, 14–19 March 2010, pp.525–533. New York: IEEE.

14. Zhang JH, Li PY, and Mao J. IPad: ID-based public auditing for the outsourced data in the standard model. *Clust Comput* 2016; 19(1): 127–138.

15. Peikert C. A decade of lattice cryptography. *Found Trend Theor Comput Sci* 2016; 10(4): 283–424.

16. Xu W, Feng D, and Liu JN. Public verifiable proof of storage protocol from lattice assumption. In: *Proceedings of the 2012 IEEE international conference on intelligent control, automatic detection and high-end equipment (ICADE 2012)*, Beijing, China, 27–29 July 2012, pp.133–137. New York: IEEE.

17. Chen L, Han LS, Jing JD, et al. A post-quantum provable data possession protocol in cloud. *Secur Commun Netw* 2013; 6(5): 658–667.

18. Yang Y, Huang Q, and Chen F. Secure cloud storage based on RLWE problem. *IEEE Access* 2018; 7: 27604–27614.

19. Cramer R, Ducas L, Peikert C, et al. Recovering short generators of principal ideals in cyclotomic rings. In: *Proceedings of the 35th annual international conference on advances in cryptology: part II (EUROCRYPT 2016)*, Vienna, 8–12 May 2016, pp.559–585. New York: Springer.

20. Wu FG, Yao W, Zhang X, et al. Lattice based signature with outsourced revocation for multimedia social networks in cloud computing. *Multimed Tool Appl* 2019; 78(3): 3511–3528.

21. Wang XM, Zhang Y, Gupta BB, et al. An identity-based signcryption on lattice without trapdoor. *J Univ Comput Sci* 2019; 25(3): 282–293.

22. Yan JH, Wang LC, Li LZ, et al. Attribute-based signcryption from lattices in the standard model. *IEEE Access* 2019; 7: 56039–56050.

23. Ruckert M and Schneider M. Estimating the security of lattice-based cryptosystems, 2010, http://eprint.iacr.org/2010/137.pdf