Journal of Information Security and Applications 39 (2018) 10–18

Contents lists available at ScienceDirect

Journal of Information Security and Applications

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

A dynamic integrity verification scheme of cloud storage data based on lattice and Bloom filter

Yunxue Yan[a], Lei Wu[a,b,*], Ge Gao[a], Hao Wang[a,b], Wenyu Xu[a]

[a] School of Information Science & Engineering, Shandong Normal University, China
[b] Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology, China

ARTICLE INFO

Article history:
Available online 2 February 2018

Keywords:
Lattice cryptography
SIS problems
Cloud storage system
Bloom filter
Quantum theory

ABSTRACT

With the development of quantum computer, making the traditional cloud storage program data integrity verification protocol is no longer safe anymore, so how to establish a safe and efficient, cost-effective cloud storage system becomes the industry's research hotspot. This paper makes improvements on the basis of the previous schemes. On the cloud storage model, we focus on the protection of user data privacy, and send the file and user signature to CSP and TPA respectively, so these methods will improve the privacy of signature information. In the cloud storage data calculation, using of lattice and Bloom filter methods, can not only resist the quantum computer attacks, but also based on the realization of dynamic integrity, improving the utilization of cloud storage space ultimately.

1. Introduction

In today's society, information technology has changed people's traditional way of life, people's dependence on information technology continues to increase in recent years. With the increase in the amount of data explosion, cloud computing came into being, greatly improving people's work efficiency. Cloud storage is based on the development of cloud computing. The birth and development of Cloud storage services provide cloud users a lot of storage space, but nevertheless, cloud storage still has some shortcomings, such as data missing, data is tampered or deleted. So cloud storage data security issues become a key step in improving cloud storage services. With the development of quantum computers, it is necessary to design a signature scheme that can resist quantum computer attacks, because the difficult problems in the traditional cryptography system can be solved in the polynomial time, so that the security of various encryption schemes is threatened.

1.1. Research background of cloud storage integrity

The concept of cloud storage is based on the development and expansion of cloud computing. As a data storage and management as the core of the cloud computing system, cloud storage for the cloud era of large data processing provides a new solution. To occupy the core position in cloud computing, cloud storage platform construction is very important. In general, the cloud storage platform [28] is divided into four layers (user access layer, application interface layer, the basic management layer, storage layer), as shown in Fig. 1. Although cloud storage has been recognized by everyone, but there are still some of the advantages and disadvantages that we need attention [27], as shown in Table 1.

With the continuous development of information technology and economic times, the arrival of digital society has become an inevitable trend. The traditional way of storage is not able to meet the current large amount of data clearly. Cloud storage came into being. Cloud storage security includes confidentiality, integrity, unforgeability. The first proposed validation data integrity is based on the integrity of RSA algorithm proposed by Ateniese et al. [5], but due to the large number of modulo exponents in RSA, so when we need to modify the data calculation efficiency will be very low. So it is not suitable for big data dynamic storage. Wang et al. [6] proposed in the cloud computing security background of the public verification method and dynamic storage, but it could not resist the quantum computer attacks.

1.2. Research background of lattice signature

The rapid development of information technology has also brought about increasingly serious security issues. With the rapid development of computer technology and network technology, people's understanding of information security more and more profound, the information security requirements of the property is also increasing, from the initial confidentiality, to the present integrity, certification, non-repudiation. As well as availability re-

* Corresponding author at: School of Information Science & Engineering, Shandong Normal University, China.
E-mail address: wulei@sdnu.edu.cn (L. Wu).

https://doi.org/10.1016/j.jisa.2018.01.004
2214-2126/

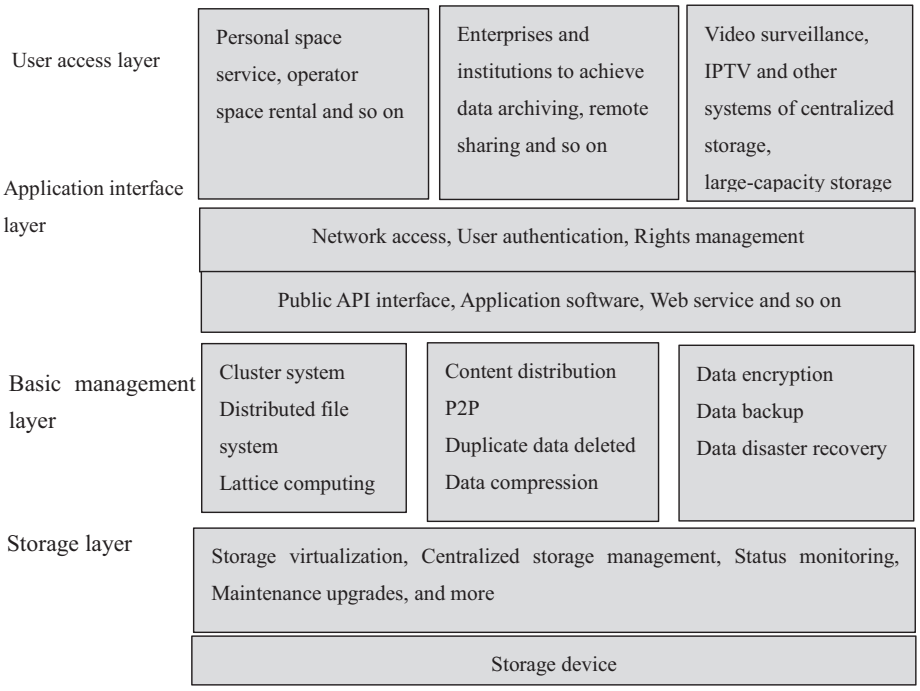

Fig. 1. Cloud storage system architecture.

Table 1
Cloud storage system advantages and disadvantages of contrast.

Advantage	Disadvantage
Low cost	The possibility of user privacy disclosure is enhanced
The management authority is clearly assigned	Data missing
Provide services on demand	Data tampering
Not subject to location constraints	Storage performance is affected by the network
Adapt to large data storage	

quirements. In order to meet people's information security requirements, we usually need to adopt the most critical and most core technology is cryptography. Early cryptography security is often based on some complex mathematical difficulties. For example, large integer decomposition problems and discrete logarithm problems. In addition, the development of elliptic curve cryptography has received a lot of attention. Zhe Liu, Johann Großschädl, Zhi Hu and so on in the new elliptic curve is very innovative put forward their own ideas, and has a better realization. For example, in the article "On Emerging Family of Elliptic Curves to Secure Internet of Things: ECC Comes of Age [29]", a study was conducted to study the calculation of the operation on a twisted Edwards curve with an effective computable internal shape, compared to conventional implementations, The number of points can be reduced by about 50%. Their design provides a variety of trade-offs and optimizations between performance and resource requirements. In the article "Elliptic Curve Cryptography with Efficiently Computable Endomorphisms and Its Hardware Implementations for the Internet of Things [30]" by defining an emerging lightweight elliptic curve family to meet the requirements of some resource-constrained devices and which has two optimized designs, High-speed version (HS) and efficient (ME) version.

Based on the traditional cryptography system is faced with the risk of quantum attack, In 1994, Shor [1] found a polynomial time algorithm for solving factorization and discrete logarithm problems in quantum computer models, which prompted cryptographers to post-quantum time's cryptographic system analysis and design research. At present, under the quantum computer model, an effective algorithm for solving difficult problems has not been found, and many problems in lattice theory have proved difficult. Therefore, the analysis and design of the cryptographic system based on the difficult problem in the lattice theory has become one of the hotspots of the "post-quantum era cryptography system". In 1996, Ajtai [2] gave a landmark conclusion on the basis of the issues of lattice problems, and proposed the possibility of constructing the cryptographic scheme based on the problem of lattice problem, and provided a new idea for constructing the new public key system. In 2008, Gentry et al. [3] proposed a lattice-based digital signature scheme, which became the basic tool for designing public key cryptography. In 2013, Wang [4] proposed a lattice-based linear homomorphic signature scheme based on GPV, but this scheme does not support data dynamic verification, Therefore, in the background of cloud storage applications, the data often need to be updated, so it does not have practical value.

1.3. Research background of Bloom filter

With the increasing amount of data, the query and modification of data become one of the core problems of cloud computing research. In the development of computer science, we often explore the question of how to improve efficiency, but often need to pay the time or space in exchange for. In 1970, Bloom Filter was proposed by Howard Bloom, which is a very long binary vector data structure. The main function is to query the collection of elements in the attribution of the problem. His obvious advantage is that it can efficiently meet the needs of the search. Bloom filter can not only represent the collection, but also can support the collection of elements to insert and query. Bloom filter Compared with the tra-

ditional hash table, in the Bloom filter hash table, an element requires only a few bits. So in 70 years after the emergence of Bloom filter, they received a warm welcome. But at the same time due to their own shortcomings and the development of the reality, bloom filter shortcomings are constantly emerging. Thus, the Bloom Filter was introduced in 2000, and the Bloom Filter was introduced in 2001. With the continuous extension and expansion of information technology, the Bloom Filter research has been deepened and various applications on Bloom Filter and improvement is also endless, I believe Bloom Filter will have a broader development. This paper mainly for Bloom Filter to achieve the dynamic integrity of the data validation.

1.4. Organization of the paper

In this paper, the present situation and research background of cloud storage and signature are summarized and improved in the existing scheme. In the new scheme, the first application is in the cloud storage background, the application of lattice signature, can better resist the quantum computer attacks, and the algorithm has been improved. Second, it combined with the Bloom Filter, in the third-party audit verification, do not need to verify a collection, but to verify the vector, and it can improve efficiency better. Structure of this paper: The first part introduces the research status of cloud storage, Bloom Filter and lattice signature. The second section introduces the related basic knowledge. In the third section, the data integrity verification scheme is described in detail. This paper analyzes the correctness and security of this scheme in the fourth section. In the fifth section, the algorithm's performance is analyzed. The sixth section leads to the conclusion of this paper.

2. Preliminaries

2.1. Lattice

Definition 1 Lattice [17].

Let b_1, b_2,...$b_n \in z^m$ be n linearly independent vectors, set $B=\{b_1, b_2, \ldots b_n\}$, Lattice $\Lambda = L(b_1, b_2, \cdots, b_n) = \{\Sigma_{i=1}^{n} x_i b_i | x_i \in Z\}$ is generated by B, and B is called the base of lattice Λ. If $m=n$, Λ is called full rank.

Definition 2 Rank and dimension [24].

In the above definition of the lattice Λ, n and m are called rank and dimension, in general, the dimension m of the lattice is greater than the rank n of the lattice. If $m < n$, then the lattice is called super rank; if $m=n$, it is called the full rank lattice; if $m > n$, it is called the reduced rank lattice.

2.2. Difficult problems on the lattice

The security of the lattice signature scheme used in this article depends on the difficulty of the SIS problem on the Lattice.

Definition 1 SIS problem.

Given $(A, t) \in SIS_{q,n,m,d} - Search$, or randomly and uniformly from $Z_q^{n\times m} \times z_q^m$, find the vector $v \in \{-d, 0, d\}^m$, so that $Av=t$ (the precondition, the SIS problem on the Lattice is defined on the integer field Z_q of the mod q).

For example, define a two-dimensional lattice Λ, as shown in Fig. 2, where the base vector is $b_1 = [{}^{1}_{1}]$, $b_2 = [{}^{1}_{-2}]$.

Definition 2 Difficulties in solving small integer solutions.

For any $m(n)=\Theta(n\log n)$, there exists a $q(n)=O(n^2\log n)$ such that the $SIS_{q,n,m\beta}$ problem for averaging the probability for an arbitrary function $\gamma(n)=\omega(n\log n)$ is at least as difficult as solving the worst case $SIVP_\gamma$ with an approximation factor of $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$.

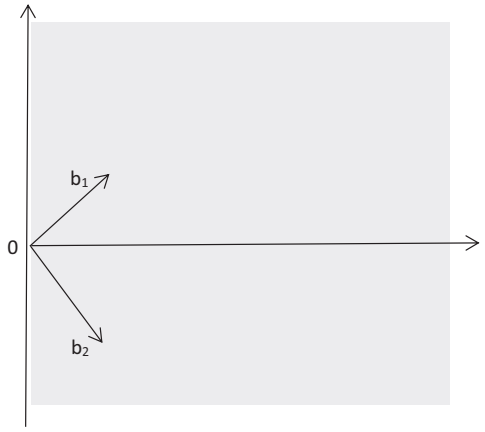

Fig. 2. Lattice definition.

2.3. Discrete normal distribution

Definition.

R^m is defined as $\sigma > 0$ as the standard deviation. The continuous normal distribution function centered on $v \in z^m$ is defined as $\rho_{v,\sigma}^m(x)$; z^m is the standard deviation from $\sigma > 0$, and $v \in Z_m$ is the center of the discrete normal distribution [16] function defined as $D_{v,\sigma}^m(x)$ among them: $\rho_{v,\sigma}^m(x) = (\frac{1}{\sqrt{2\pi\sigma^2}})^{m_e} {}^{\frac{-\|x-v\|^2}{2\sigma^2}}$. In particular, when $v=0$, $\rho_{v,\sigma}^m$ and $D_{v,\sigma}^m$ can be abbreviated as ρ_σ^m and D_σ^m, respectively.

Lemma for arbitrary $\sigma > 0$ and positive integers m, have

(1) $\Pr[x \leftarrow D_\sigma^1 : |x| > 12\sigma] < 2^{-100}$;
(2) $\Pr[x \leftarrow D_\sigma^m : \|x\| > 2\sigma\sqrt{m}] < 2^{-m}$;

2.4. Gaussian function and Gaussian distribution

Definition 1 R^n on the $s > 0$ as a parameter, $c \in R^m$ as the center of the Gaussian function is defined as

$$\forall x \in R^m,\ \rho_{s,c}(x) = \exp(-\pi \|x-c\|^2/s^2)$$

The total measure associated with the Gaussion function is $\int_{x\in R^n} \rho_{s,c}(x) = \exp(-\pi \|x-c\|^2/s^2)$.

Thus, the continuous Gaussian distribution with c as the center and the parameter $s > 0$ is

$$\forall x \in R^m,\ D_{s,c}(x) = \rho_{s,c}(x)/s^n$$

Here when c and s are not explicitly stated, respectively, defaults to 0 and 1.

Definition 2: Discrete Gaussian distribution on the lattice, for any vector c, the real $s > 0$ and the lattice, the discrete Gaussian distribution on the defined Λ, Λ is:

$$\forall x \in \Lambda,\ D_{\Lambda,s,c}(x) = \frac{D_{s,c}(x)}{D_{s,c}(\Lambda)} = \frac{\rho_{s,c}(x)}{\rho_{s,c}(\Lambda)}$$

Among them: $D_{s,c}(\Lambda) = \Sigma_{x\in\Lambda} D_{s,c}(x)$, $\rho_{s,c}(\Lambda) = \Sigma_{x\in\Lambda}\rho_{s,c}(x)$

For a sufficiently large parameter $s > 0$, the discrete Gaussian distribution $D_{\Lambda,s,c}(x)$ and the continuous Gaussian distribution $D_{s,c}(x)$ on the lattice are very similar in many respects. For example, the mean value of the vector obeying the distribution $D_{\Lambda,s,c}(x)$ is very close to the center c, and the expected value of the square of the distance of the vector is very close to $s^2n/2$. For the vector that follows the distribution $D_{s,c}(x)$, the two values are exactly c and $s^2n/2$. Then, define a lattice parameter, smooth the parameters, so that the parameters $s > 0$ to achieve when the two distribution is very close.

Define 3 Smooth parameters.

For any n-dimensional lattice Λ and real numbers $\varepsilon > 0$, Define its smoothing parameter $\eta_\varepsilon(\Lambda)$ as the minimum s, making $\rho_{1/s}(\Lambda^*\backslash\{0\}) \le \varepsilon$ among them:

$$\rho_{1/s}(\Lambda^*\backslash\{0\}) = \sum_{x\in\Lambda*\backslash\{0\}} \rho_{1/s,0}(x).$$

The following lemma gives the upper and lower bounds of smoothing parameters

Lemma 1. *For any n-dimensional lattice, have* $\eta_\varepsilon(\Lambda) \le \sqrt{n}/\lambda_1(\Lambda^*)$, *among them* $\varepsilon = 2^{-n}$.

Lemma 2. *For any n-dimensional lattice and real number* $\varepsilon > 0$, *have:*

$$\eta_\varepsilon(\Lambda) \le \sqrt{\ln(2n(1+1/\varepsilon))/\pi}.\lambda_n(\Lambda)$$

Lemma 3. *Making* Λ *is a n-dimensional lattice, for any* $\varepsilon \in (0.1)$, $s \ge \eta_\varepsilon(\Lambda)$ *and* $c \in R^n$, *have:*

$$\rho_{s,c}(\Lambda) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, 1\right] \cdot \rho_s(\Lambda)$$

2.5. Reject the sampling theorem

If f and g are probabilistic distribution functions, there exists $M \in R$ for arbitrary x satisfies $f(x)$. if the sampled z is taken from the distribution g and the probability $\Pr = f(z)/(Mg(z)$ (Z, c), then this pair (z, c) can be seen as an instance of the rejection of the sample [7], while the output of the distribution is f, output an instance (z, c). the number of times required is M.

Lemma Reject the sampling theorem basic properties:

(1) $\Pr[\|z\|] > \omega(\sigma\sqrt{\log m}); z\in_R D^1_\sigma = 2^{-\omega\sqrt{\log m}}$, specifically, $\Pr[\|z\| > 12\sigma; z\in_R D^1_\sigma] < 2^{-100}$;
(2) $\forall z \in z^m$, $\sigma \ge \sqrt{\log(3m)}$, have, $D^m_{v,\sigma} \le 2^{-m+1}$;
(3) $\Pr[\|z\| > 2\sigma\sqrt{m}; z\in_R D^m_\sigma] < 2^{-m}$.

2.6. Bloom filter

Definition 1 [23]**.** The Bloom Filter of the element indicates that the vector is denoted as $BF^{k,m(x)}$, $BF^{k,m(x)}$ for any element x in the complete set U through the k hash functions, $\text{hash}_i(x)] = 1$ $(1 \le i \le k)$, the element-to-Bloom filter vector is mapped $x \xrightarrow{k,m} BF^{k,m}(x)$.

Definition 2. The Bloom filter of the set represents that for all the elements in the complete subset U, all elements in $S = \{s_1, s_2, \ldots s_n\}$ are passed through K. The hash function is expressed as the length of m-bit Bloom filter vector, and the vector is called $BF^{k,m(S)}$. The mapping process to the Bloom filter vector is $S \xrightarrow{k,m} BF^{k,m}(S)$.

Definition 3. The homologous Bloom filter uses the same k hash functions for any subset S in the complete set U, mapped to the Bloom filter vector of the same length m, and a class of Bloom filter called the homologous bloom filter. Denoted as $BF^{k,m}$, $BF^{k_0,m_0} = \{BF|\forall S \in U, S \xrightarrow{k=k_0,m=m_0} BF^{k_0,m_0}(S)\}$.

Definition 4. Count-type Bloom filter is introduced due to the limitations of the standard Bloom filter functionality, such as dynamic data support. as shown in Fig. 3.

3. Data integrity verification scheme

3.1. Scheme model

In this paper, the data integrity verification scheme includes three entity participants: Cloud Server Provider (CSP), User (User), and Third Party Auditor (TPA) [12]. In the cloud storage structure,

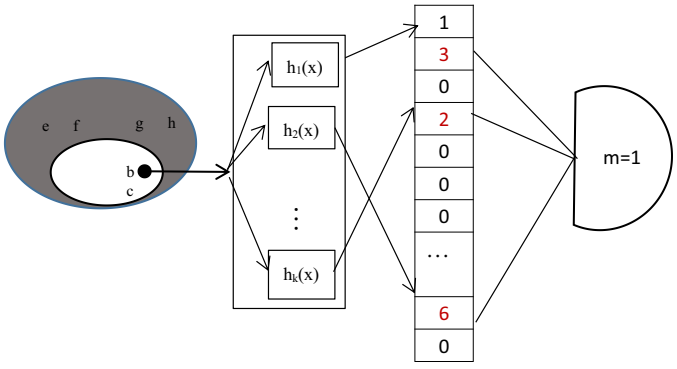

Fig. 3. Counting Bloom filter principle.

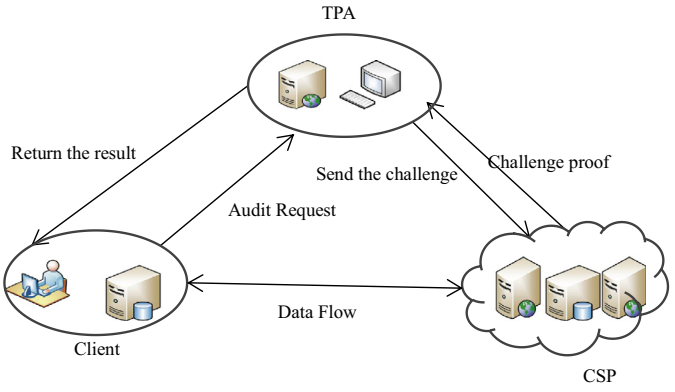

Fig. 4. Cloud storage model.

the user needs to register their own account and upload their own data to the cloud server. Due to the hardware, data validation conditions and operational complexity constraints, it is difficult for users to verify the integrity of the data. Hence, it is necessary to introduce a third party for integrity verification. The specific process is shown in Fig. 4.

User: After the user registers the account, uploading the file to the cloud server. The right to verify the integrity of data entrusted to the TPA, so in the cloud storage model, the user can get their own data from the remote, and it will own a lot of data stored in the cloud, but also do not have to worry about local storage's management.

Cloud Service Provider (CSP): Cloud storage service providers store a large amount of data from users, providing many different types of cloud servers, providing users with storage or computing services that enable users to upload and download data anytime, anywhere. So it has a strong computing power and a large storage space, but also existing data loss and forged phenomenon, but for the user, may have concealed it.

Third Party Audit (TPA): In the cloud storage integrity verification scheme, third party auditing plays an important role. In this scheme, it is assumed that TPA is an unbiased, credible entity. In the integrity verification process, it replaces the user to exercise the integrity of the right to verify. The third party audits challenge the cloud storage service provider based on the user's audit request, and then receive the evidence sent back by the cloud storage service provider to verify the integrity of the data and feed the results back to the user.

3.2. Bloom filter function introduction

In this section, we introduce the basic functions of Bloom filter and Counting bloom filter, including query and insert. And gives its corresponding code, as shown in Fig. 5.

Algorithm 1 Insert(element)

Require: element is not null

1:Active BF ←Get Active Standard BF()
2:**if** Active BF is null **then**
3: Active BF ←Create Standard BF(m,k)
4: **Add** Active BF to this dynamic bloom filter
5: s ←s+1
6:**for** i=1 to k **do**
7: Active BF[hash$_i$[(element)] ←1
8: Active BF.nr ←Active BF.nr+1

Get Active Standard BF()

1: **for** j=1to s **do**
2: if Standard BF$_{j.nr}$<n$_0$ **then**
3: **Return** Standard BF$_j$
4:**Return** null

Algorithm 2 Query(element)

Require: element is not null

1:f**or** i=1 to s **do**
2: counter ←0
3: **for** j=1 to k **do**
4: **if** Standard BF$_i$[hash$_j$[(element)]=0 **then**
5: **break**
6: **else**
7: counter ←counter+1
8: **if** counter=k **then**
9: **Return true**
10:**Return false**

Fig. 5. Bloom filter related algorithms.

Bloom Filter is a data structure with high efficiency. Bloom Filter uses vector V to represent the set $S=\{s_1, s_2 \ldots s_n\}$ of n members, where the initial value of vector V is set to 0. The size of the vector is m bits. The elements in set S are mapped to vector V using k independent hash functions. For the general Bloom filter, his main function is to determine whether the element belongs to this collection. If you want to determine whether x belongs to set S. First, let each element s belong to V, let $V[h_i(s)]=1$, $i \in \{1,2, \ldots k\}$;Then, checking whether $V[h_i(x)]=1$ for each $i \in \{1,2, \ldots k\}$, Finally, if there is not a case of 1, then x does not belong to the set S, if all 1, then the element x belongs to the set S. When the total value of 1, we judge the element x belongs to the set S, sometimes interpretation is wrong. This is the false rate about Bloom filter. In practical application, we can choose the appropriate bloom filter by the number of set elements and the expected false positives [8].

Thus, as shown in Figs. 6 and 7, it is not difficult to find that the Bloom filter essentially maps the elements in the set to the corresponding positions by K hash functions. Bloom filter can be truly efficient and simple to carry out the query elements.

3.3. Protocol implementation

The scheme proposed in this paper consists of five parts: Setup () Encode () Challenge () Prove () Verify (). Algorithm is described as follows [18].

Setup(): Assuming that the hash function $H:\{0,1\}^* \rightarrow \{v: v \in \{-1, 0, 1\}^k, \|v\| \leq k\}$ in the lattice signature scheme is the random oracle model, set its output 100bit;The user selects the m-dimensional vector y in the discrete uniform distribution D_σ^m, where σ is the standard deviation of D_σ^m. From $\{-d, 0, d\}$ randomly selected integer configured $m \times k$ dimensional matrix S as the user's private key; The matrix $A(n \times m$ dimensional) is constructed by randomly selecting integers from Z_q; $T = AS \in Z_q^{m \times k}$.$\{A, T\}$ as the public key.

Encode(): Based on the above parameters, the user's file is signed by using SIS difficult problem [7]. Specific steps can be divided into the following five parts.

Step 1: The m-dimensional vector y is randomly selected from the distribution D_σ^m;

Step 2: The user divides the file F into L data blocks, $F=\{\mu_1, \mu_2,\ldots,\mu_i\}$, $\mu_l \in Z_q^m$;

Step 3: Calculate $e_i = H(Ay, u_i)$, where u_i is the message to be signed; calculate $z_i = Se_i + y$ and put e_i into the signature set $\phi=\{e_1, e_2, \ldots e_i\}$.

Step 4: The signature result (z_i, e_i) is outputted in succession $\Pr = \min(\frac{D_\sigma^m(z_i)}{MD_{Sc.\sigma}^m(z_i)}, 1)$, Where the output results are used the rejected sampling theorem.

Step 5: After receiving the signature pair (z_i, e_i), calculate $\|z_i\|$. If $\|z_i\| \leq 2\sigma\sqrt{m}$ and $e_i = H(Az_i - Te_i, \mu_i)$ is true, the user sends the original signature to the TPA and the file to the CSP.

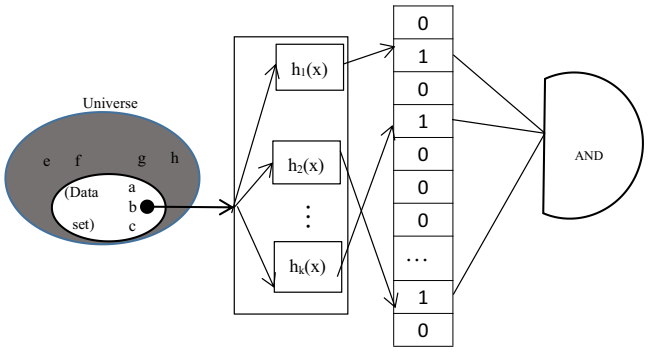

Fig. 6. Standard Bloom filter.

TPA builds MHT according to the signature set ϕ, the value of the root node is obtained by the anti-collision hash function $H(\cdot): \{0,1\}^* \rightarrow z_q^n$. According to the root node structure filter, therefore, we assume that the root node is $\{h_{R_1}, h_{R_2}, \ldots h_{R_n}\}$. Then:

Step 1: Constructs an array $X_{[n]}$ of length n, initializing each bit of the array to zero, $X_{[j]}=0$ $(0 \leq j \leq n)$

Step 2: Define k different and independent of the hash function, $H_1(x)$, $H_2(x) \ldots H_k(x)$.

Step 3: The root node h_{R_1}, h_{R_2}, $\cdots$ h_{R_n} as the K hash function input, the details are calculated as follows:

$$\begin{array}{ll} H_1(h_{R_1}) = a & X_{[a]} = 1 \\ H_2(h_{R_1}) = b & X_{[b]} = 1 \\ \cdots & \cdots \\ H_K(h_{R_1}) = j & X_{[j]} = 1 \end{array} \quad \rightarrow$$

$$\begin{array}{ll} H_1(h_{R_2}) = e & X_{[e]} = 1 \\ H_2(h_{R_2}) = f & X_{[f]} = 1 \\ \cdots & \cdots \\ H_k(h_{R_3}) = g & X_{[g]} = 1 \end{array} \quad \rightarrow$$

After all the nodes have completed the above operations, the result of the array can only be $X_i = 0 \| 1$.

Challenge (): The user sends an audit request AudiQuest (i) to the TPA. After the TPA authenticates the user's identity, the TPA sends chal $\{\mu_i\}$ to the CSP.

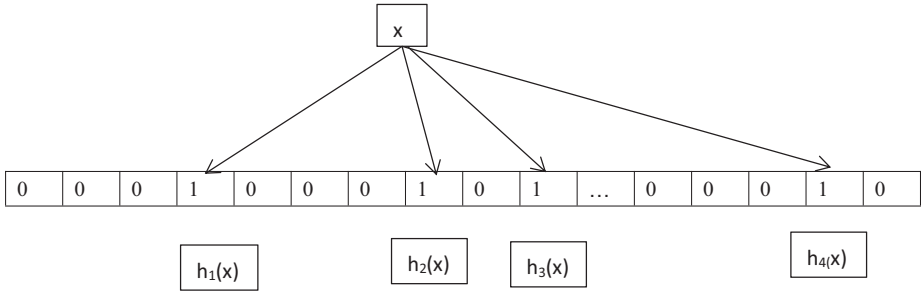

Fig. 7. Insertion of element Bloom filter.

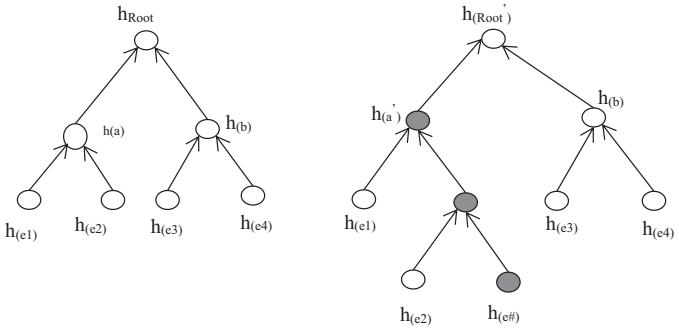

Fig. 8. MHT performs an insert operation.

Prove (): CSP through authentication, It can accept the challenge from the TPA. Cloud Storage Provider generates the corresponding file signature $e_{\#}$ sent to the TPA according to the challenge.

Verify (): The TPA calculates the corresponding h'_{Root} according to the signature sent by the CSP. h'_{Root} is inserted as a new element into the Bloom Filter, if all $X[H(h'_{root})] = 1$, the data block μ_i integrity is passed, otherwise the data block μ_i is modified.

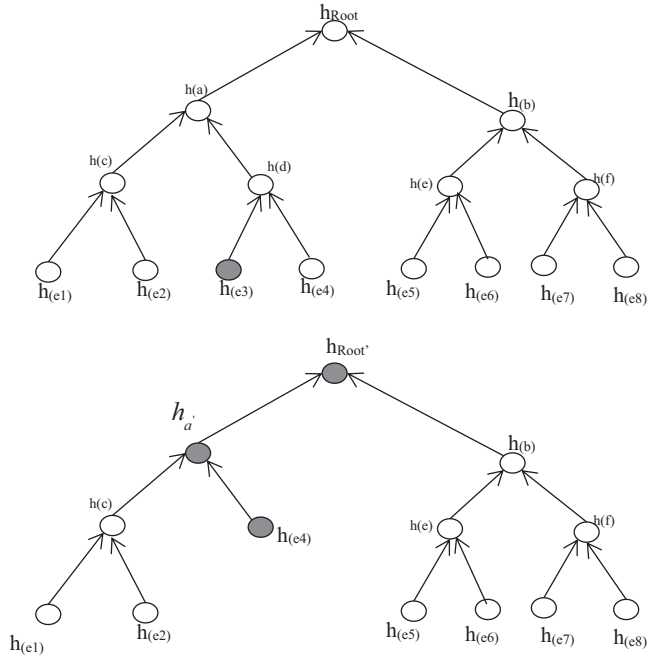

Fig. 9. MHT performs a delete operation.

3.4. Dynamic operation

As cloud computing requirements continue to increase, data integrity requirements are getting higher and higher. This scheme supports static data and dynamic data integrity verification. Specific operations for the data include: data insertion, deletion, update. In this scheme, a technique similar to that of [10] is used to introduce the hash about the Merkle hash tree. This section focuses on the use of Counting Bloom Filter [15] to enable dynamic data operation.

In this section, the operation of the data can be divided into two steps, first of all, we need to re-modify our root value. As shown in Figs. 8 and 9:

For a count-type Bloom filter, each bit of a one-dimensional array is expanded by a small counter. First of all, when data is inserted, the $H_k(h_{Root'})$ is calculated based on the hash value of the new root node provided by the Merkle hash tree. Second, let $X[H_k(h_{Root})]$ is decremented by 1. Let $X[H_k(h_{Root'})]$ is incremented by 1. Therefore, after the value of the root node is filtered, the value of each bit in this array is not limited to 0 or 1 which compared to the hash value of the normal Bloom Filter. As shown in Fig. 10.

4. Security analysis

In the proof of the integrity verification scheme, in order to determine the integrity of the data in the cloud, the verifier needs to periodically initiate a verification request for the data in the cloud. According to the existing integrity verification scheme [20–22]. The security model that defines the data integrity verification scheme based on the lattice is that there is no polynomial adversary in the verification process, allowing the verifier to receive the evidence that he generates in a non-negligible advantage.

The cloud integrity verification scheme based on the lattice and bloom filter proposed in this paper, it can be formally described as follows: Adversary A can produce a set of forged proofs in a polynomial time range with a non-negligible advantage, and the proof can pass verification, there are challenges on the basis of this can solve the problem on the lattice.

In this section, through the security analysis [26], it can prove that the scheme can meet the basic requirements of integrity, privacy, security, confidentiality. The integrity check process is shown in Fig. 11.

4.1. Privacy

Privacy is the TPA that plays the third party public audit role in the cloud storage integrity verification model based on the lattice and Bloom filter proposed in this paper. It cannot be used in the whole process of data integrity verification of the original data, and access to the user's data and the user's secret information, so this guarantees the privacy of the user's data.

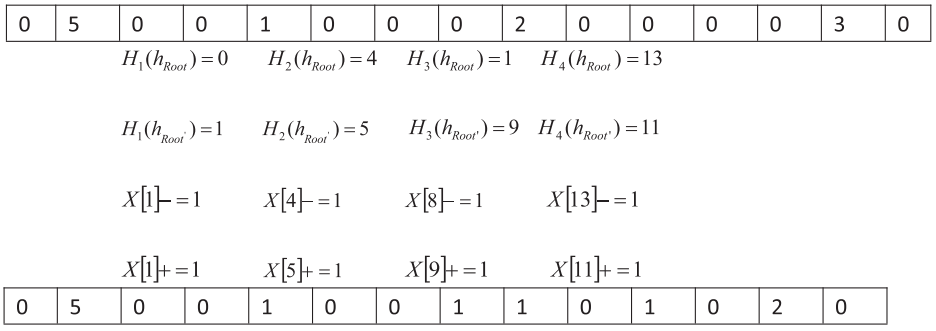

Fig. 10. Counting Bloom Filter for insert operation.

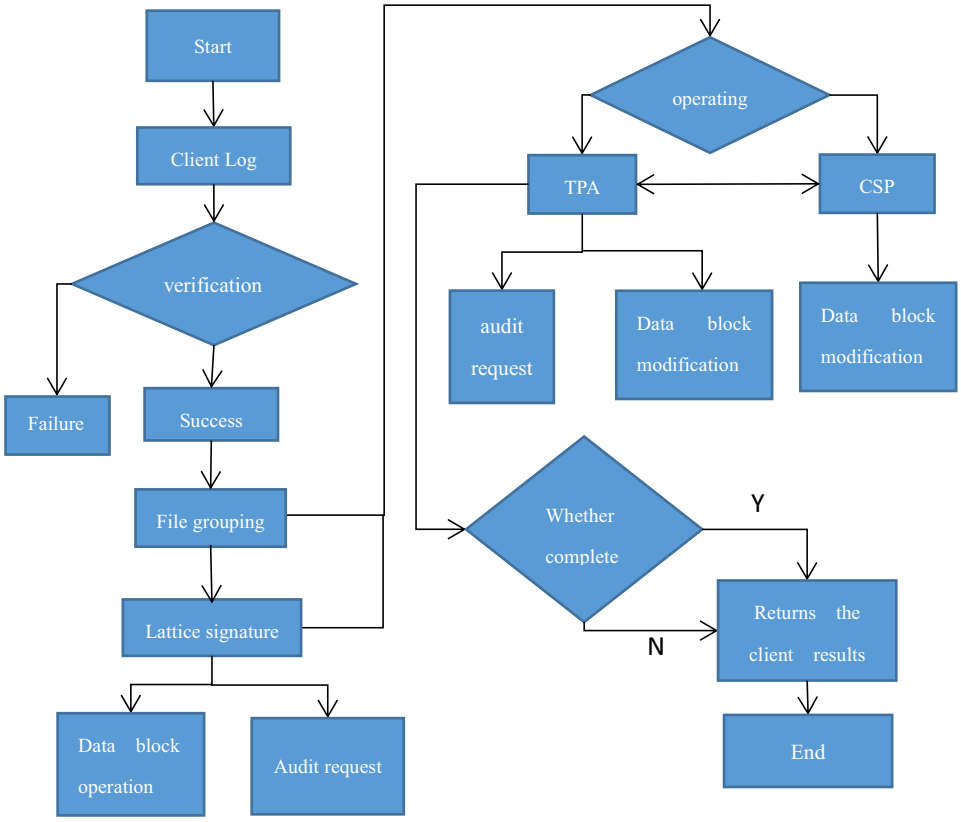

Fig. 11. Integrity verification process.

In this paper, the Lattice signature scheme is used to reject the sampling technique in mathematical knowledge. First, comparing the Gaussian original image sampling algorithm, the efficiency is improved. Secondly, when we design a signature scheme, denial of sampling technology is to eliminate the dependence of the signature on the private key S. Hence, it can better protect the privacy of the algorithm. First of all, the signature algorithm extracts the vector y in the discrete gaussian distribution D_σ^m, then calculating $e_i = H(Ay \bmod q, \mu_i)$, and finally calculating $z_i = Sc_i + y$. Then, outputting signature result (z_i, c_i). In order to get samples from the distribution $D_{Sc,\sigma}^m$. Thus, using a reject sample, it is necessary to find a positive real number M, which satisfies D_σ^m for all samples x according to $D_\sigma^m(x) \le M \cdot D_{Sc,\sigma}^m(x)$.

4.2. Correctness

Correctness means that in the new lattice signature scheme proposed in this paper, the valid signature of the algorithm output through the algorithm can be verified.

Prove: On the one hand,

$$\begin{aligned} &A \cdot z_i - T \cdot e_i \\ &= A \cdot (S \cdot e_i + y) - T \cdot e_i \\ &= A \cdot S \cdot e_i + A \cdot y - T \cdot e_i \\ &= A \cdot y \bmod q \end{aligned}$$

On the other hand, due to z_i the distribution D_σ^m of satisfaction, it can be drawn: $\|z_i\| \le 2\sigma\sqrt{m}$.

4.3. Unforgeability

Unforgeability means that in the scheme proposed in this paper, an attacker cannot obtain the user's signature for the acquired fragment information to forge the user's signature, thereby obtaining the data about the user. In this paper, we mainly use the problem of solving short integers based on lattice.

Assuming that an adversary is in the process of query, after obtaining A and T, the signature of the scheme can be breached in an unignorable probability that the adversary can forge an effective

Table 2
Analysis of scheme characteristics.

	The scheme in Ref. [10]	The scheme in Ref. [11]	The scheme in Ref. [14]	Our scheme
Integrity verification	Yes	Yes	Yes	Yes
Dynamic update	Yes	Yes	Yes	Yes
Third party audit	No	Yes	Yes	Yes
Whether the signature algorithm can resist quantum computer attacks	Yes	No	Yes	Yes
Whether the third party can obtain user files	No	Yes	Yes	No

Table 3
Analysis of scheme efficiency.

	Whether it is a random oracle model	Whether it is sampled	Public key length	Private key length	Signature length
The scheme in Ref. [9]	No	Yes	$(\|\mu\|+1)nm\log q$	$m^2\log q$	$m(\|\mu\|/2+1)\log q$
The scheme in Ref. [13]	No	Yes	$(nm+(\|\mu\|+2)n^2k+n)\log q$	$mnk\log q$	$m+2nk\log q$
The scheme in Ref. [16]	Yes	No	$2nm\log q$	$m^2\log 3$	$m\log(12\sigma)$
Our scheme	Yes	No	$m\log(12\sigma)$	$mk\log(1+2d)$	$m\log(12\sigma)$

signature. Then, we can construct a probabilistic polynomial time algorithm to solve the problem of the small integer solution by using these queries. That is, we can use this algorithm to solve the vector v in SIS problem, so that $AV=t$, so there is no such probability polynomial algorithm, which can prove that the signature is not falsifiability.

4.4. Bloom filter security analysis

Bloom Filter has some excellent features in verifying data integrity, which is not available in other schemes. First, Bloom Filter generates a generation vector and auxiliary update vector for each data block during operation. In the data search and data update, it is helpful to improve the efficiency of the operation and the storage efficiency of the ciphertext. Secondly, Bloom Filter also uses the unidirectionality of the hash function, there is a certain probability of fault tolerance. For details, please refer to the literature [15]. Finally, the scheme chooses Bloom Filter because it can resist adversary attacks on some aspects.

A detailed analysis is as follows: (1) To resist the known part of the plaintext or ciphertext corresponding to the attack. In the process of this scheme encryption, the file is divided, the signature generated by each different data block is independent. So even if the attacker obtains some of the ciphertext information in the encryption process, but also it still can't obtain sensitive information; (2) To prevent the data block part of the tampered or replacement attacks. In Bloom Filter, each data block has a corresponding authentication device, when the data block is maliciously modified, it can return the corresponding results through the authenticator . We can carry out integrity checks through the authenticator, which can effectively resist the plaintext / ciphertext replacement attack. (3) Data's security. Bloom Filter uses the hash function to generate authenticator' data, and uses the characteristics of the hash function, and combines with lattice signature to protect the data in this scheme. On the one hand, the probability of certification failure is small, on the other hand, when the cloud storage system model is designed, the user's data information can be hidden well.

5. Performance analysis

This section mainly analyzes the performance of the scheme, including the analysis of the characteristics of the scheme, the data security analysis of the scheme and the efficiency comparison of the scheme [25]. Among them, through the comparison between the schemes, it is hoped that the advantages and disadvantages of different schemes can be better understood, so as to better guide the next step in the program.

5.1. Algorithm characteristics analysis

Data integrity verification program, for a variety of application background has a corresponding to different programs. This section compares the characteristics of the different integrity verification schemes, including whether the data can be dynamically updated, whether it supports third parties for open integrity verification, whether it can resist quantum computing attacks, and protect the privacy of user data, Whether the third party audit can obtain the user's original file. So that we can better compare the differences between the programs, as shown in Table 2. In the proposed solution, TPA and CSP have different rights, and the user's data and signature are sent to CSP and TPA respectively, In the proposed cloud storage model, Which can be better to hide the privacy of data.

5.2. Algorithm efficiency analysis

With the increase in the amount of data explosion, cloud storage technology, one of the core issues is to improve the efficiency of cloud computing. As shown in Table 3, the efficiency of the algorithm for different scenarios is compared, including whether to sample, whether to use the random oracle model, and the public key length, private key length, signature length. In the scheme proposed in this paper, we reject the sampling technique and avoid the use of expensive Gaussian sampling in the random oracle model, which not only improves the efficiency, but also maintains the security of the data. Finally, on the signature length, Cryptography is signed with a shorter signature length.

6. Conclusion

Data security is one of the most concerned about cloud users, but also a key step in restricting the development of the cloud storage technology. In our scheme, through the combination of lattice signature and Bloom Filter theory, not only to achieve the resistance of quantum computer [19] attacks, and because the Lattice and Bloom Filter are operated in the vector space, thereby enhancing the cloud storage space utilization. With the increasing number of data in cloud storage, how to make more efficient use of cloud storage space, how to protect the user's privacy under the premise of improving the efficiency of TPA verification, will be our next step. From the current view, cloud storage data security issues still have great research value and a very broad space for development. If we can explore these issues in depth, I believe it will be better to promote the development of cloud storage platform and make cryptography system more perfect.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61602287, 61672330).

References

[1] Shor P. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th annual symposium on foundations of computer science, FOCS1994. IEEE; 1994. p. 124–34.
[2] Ajtai M. Generating hard instances of lattice problems (extend abstract). Proceedings of STOC, 996:99–108.
[3] Gentry C, Peikert C, Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions. In: DBLP; 2008. p. 197–206.
[4] Wang F, Hu Y, Wang B. Lattice-based linearly homomorphic signature scheme over binary field. Sci China Inf Sci 2013;56(11):1–9.
[5] Ateniese G, Burns R, Curtmola R, et al. Provable data possession at untrusted stores. In: ACM conference on computer and communications security. ACM; 2007. p. 598–609.
[6] Wang Q, Wang C, Ren K, et al. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans Parallel Distrib Syst 2011;22(5):847–59.
[7] Cao J, Yang Y, Zichen LI, et al. Lattice signature and its application based on small integer solution problem. J Comput Appl 2014.
[8] Xiao MZ. A survey on Bloom filters and its applications. Comput Sci 2004.
[9] Wang F, Hu Y, Jia Y. Lattice-based signature scheme in the standard model. J. Xidian University 2012.
[10] Li X, Ye TM. Based on other lattice with large data storage integrity verification scheme. Inf Secur 2014(4):46–50.
[11] Hu D-M, Yu X. A verification method of verification of dynamic cloud storage data integrity based on homomorphism labels. Appl Res Comput 2014;31(5):1362–5.
[12] Han K, Li Q, Deng Z. Security and efficiency data sharing scheme for cloud storage. Chaos Solitons Fractals Interdiscip J Nonlinear Sci Nonequilib Complex Phenom 2016;86:107–16.
[13] Micciancio D, Peikert C. Trapdoors for lattices: simpler, tighter, faster, smaller. In: International conference on the theory and applications of cryptographic techniques. Springer; 2012. p. 700–18.
[14] Tan S, He L, Chen Z, Jia Y, et al. A method of provable data integrity based on lattice in cloud storage. Comput Res Develop 2015;52(8):1862–72.
[15] Ren HQ, Jian-Zhu LU, Jiao-Yang XU. Bloom Filter-based field authentication scheme for encrypted database. Comput Eng Des 2011;32(3):818–21.
[16] Zhang XS, Liu ZH, Science SO, et al. Non-trapdoors lattice signature scheme with message recovery. Comput Sci 2014.
[17] Zhao X, Wang X, Xu H, et al. Cloud data integrity checking protocol from lattice. Int J High Perform Comput Networking 2015;8(2):167.
[18] Hongwei Wenming. Public proof of cloud storage from lattice assumption. Chin J Electron 2014;23(1):186–90.
[19] Zhang Y, Liu Q, Tang C, et al. A lattice-based designated verifier signature for cloud computing. Int J High Perform Comput Networking 2015;8(2):135–43.
[20] Shacham H, Waters B. Compact proofs of retrievability. Advances in cryptology – ASIACRYPT 2008 Berlin Heidelberg. Springer; 2008.
[21] Wang C, Wang Q, Ren K, et al. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. INFOCOM, 2010 Proceedings IEEE, 2010, 62(2):525–533.
[22] Wang Q, Wang C, Ren K, et al. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans Parallel Distrib Syst 2011;22(5):847–59.
[23] Guo D, Wu J, Chen H, et al. Theory and network applications of dynamic bloom filters. IEEE INFOCOM 2006. IEEE international conference on computer communications. IEEE, 2006:1–12.
[24] White TD, Suwa G, Asfaw B. Complexity of lattice problems: a cryptographic perspective. Siam J Comput 2002;671(6495):x,220.
[25] Liu C, Yang C, Zhang X, et al. External integrity verification for outsourced big data in cloud and IoT. Future Gener Comput Syst 2015;49(C):58–67.
[26] Cao L, He W, Liu Y, et al. An integrity verification scheme of completeness and zero-knowledge for multi-Cloud storage. Int J Commun Syst 2017(1):e3324.
[27] Kamara S, Lauter K. Cryptographic cloud storage. In: International conference on financial cryptograpy and data security. Springer-Verlag; 2010. p. 136–49.
[28] Zeng W, Zhao Y, Ou K, et al. Research on cloud storage architecture and key technologies. In: International conference on interaction sciences: information technology, culture and human. ACM; 2009. p. 1044–8.
[29] Liu Z, Huang X, Hu Z, et al. On emerging family of elliptic curves to secure internet of things: ECC comes of age. IEEE Trans Dependable Secure Comput 2017;14(3):237–48.
[30] Liu Z, Großschädl J, Hu Z, et al. Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things. IEEE Trans Comput 2017;66(5):773–85.