Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Security Analysis and Signature Algorithm Design of Gegami's Identity-Based Chameleon Signature Scheme under Quantum Attack Resistance

Guoren Xiong[1], Daofeng Li[1,✉]

[1] *Computer and Electronics Information School of Guangxi University, Nanning, Guangxi, 530004, China*

ABSTRACT

Traditional digital signatures are often publicly verifiable, and in certain applications with privacy preservation requirements, the signer does not want the sensitive information it signed to be re-delivered by a dishonest verifier. Aiming at the problem that traditional chameleon signatures (CS) cannot resist quantum computer attacks, this paper proposes a lattice-based authentication CS scheme. Based on the analysis of the lattice difficulty problem and the security vulnerability of the CS scheme, it is pointed out that it does not satisfy the third-party unforgeability and the signer rejectability, and a new lattice-based identity CS scheme is established, which is verified under the stochastic predicate machine model, and the storage and transmission efficiency of the scheme is analyzed. The results show that the newly designed identity-based CS scheme on the lattice can effectively resist quantum computer attacks, can sign messages of arbitrary length, and possesses more lightweight storage and transmission efficiency. The optimized chameleon signature scheme has better security and also provides a new solution for digital signatures to resist quantum computer attacks.

Keywords: lattice; chameleon signature; anti-quantum attack; security; signature algorithm

1. Introduction

A traditional digital signature should be publicly verifiable, transmittable and unforgeable, i.e., any user is able to verify and believe (or reject) the authenticity of a document and the identity of its signer [1-2]. However, these properties do not satisfy certain special scenarios aimed at preventing the dissemination of document contents by the verifier. In 1989, non-repudiation signatures were proposed, which require the verifier to send an online request to the signer before attempting verification and inevitably involve heavy zero-knowledge proof systems, resulting in generally inefficient signatures [3]. In 1996, designated verifier signatures were proposed, which are a method

✉ Corresponding author.
E-mail address: xgren0717@163.com (D. Li).
Received 25 March 2024; Revised 05 June 2024; Accepted 15 December 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-372

by which the signer and the designated verifier have the same level of signature authority, and any third party other than the two cannot confirm the true generator of the signature [4]. Therefore, no means of arbitration can be provided when both parties dispute the authenticity of the signature. In 1998, the chameleon signature was proposed to more cleverly solve the problem of the designated verifier's secondary transmission of the signed document [5]. In 1984, the identity-based cryptosystem was proposed, where the user's public key can be publicly computed based on the identity identity marking, and the corresponding private key is dispatched by the private key generator for the user based on the identity, and therefore no longer rely on digital certificates of traditional public key cryptosystems [6]. Combining the advantages of identity-based cryptosystems and the security features of chameleon signatures, Identity-Based Chameleon Signature (IBCS) is particularly suitable for application scenarios such as cloud healthcare systems, copyright protection, e-bidding and online/offline signatures [7-9].

Cryptographic regimes based on classical number-theoretic puzzles are not resistant to attacks from quantum computers, which means that most existing cryptographic schemes in the post-quantum era will be broken in polynomial time. Lattice cryptosystems with resistance to quantum computing attacks stand out thanks to the lightweight algebraic operations on the lattice and the existence of an average-case to worst-case security statute for the difficulty of some difficult problems on the lattice [10-11]. The IBCS scheme on the lattice, on the other hand, avoids the security loophole that arbitrary third parties can forge signatures and the signer cannot reject signatures forged by the specified verifier, and reduces the transmission overhead of the final signature from a square to a linear level [12]. However, the validity and security of traditional digital signatures have been drastically reduced under quantum attacks, which seriously threatens this financial and governmental business and other industries [13-14]. At the same time, the anti-quantum cipher also enters the transition stage of standardization, the security based on the lattice cipher suffers from threats, and there are also problems such as the shortest vector problem and the nearest vector problem, and there is no algorithm that can solve these problems efficiently among the currently known quantum algorithms [15-17]. Based on this, analyzing and designing security and signature algorithms for lattice-based IBCS under anti-quantum attacks can help provide guidance for innovative research and security proofs of lattice-based IBCS.

In this paper, we take the definition of lattice as the entry point, analyze the basic principles of chameleon signature and hash function, and establish the identity-based chameleon hash function construction method on lattice. Then the identity-based CS scheme is designed based on the identity-based CS scheme on the lattice from the perspective of improving the quantum attack resistance of the signature scheme. For the effectiveness of the scheme, the verification analysis is carried out in terms of unforgeability and quantum attack resistance, and the effectiveness of the chameleon hash function construction method in this paper is explored. Finally, the feasibility of the identity-based CS scheme on Grid is analyzed in terms of scheme efficiency and storage computation overhead.

2. Preparatory knowledge

The rise of the digital era makes more and more information need to be transmitted, stored and processed in the digital environment, and cryptography technology becomes particularly important as an important means of protecting information security. However, with the rapid development of quantum computing technology, the security of traditional public key cryptography algorithms has been threatened. Lattice cipher has been widely emphasized and researched as a public key cryptography system that is resistant to quantum attacks. In addition, lattice cipher has the advantage of worst-case to average-case normalization. By generalizing the worst-case security to the difficult problem on the lattice, the lattice-based cryptographic scheme is guaranteed to have stronger security.

2.1. Problems of gridding and gridding difficulties

2.1.1. Definition of grid. The lattice L generated by V is defined as follows:

$$\mathrm{L}(V)=\{x_1v_1+x_2v_2+\cdots+x_nv_n : x_1,x_2,\cdots,x_n\in\mathbb{R}^m\} \tag{1}$$

where $V\in\mathbb{R}^m$ is a matrix, generated by n linearly independent vectors $\{v_1,\cdots,v_n\}$ and $x_n\in\mathbb{R}^m$ are the coefficients. Furthermore $v_1,\cdots,v_n$ is the basis of the lattice $\mathrm{L}(V)$ and n is the dimension of the lattice.

Let Λ be a (x^n+1)-cyclic lattice, and if there is a $(a_1,a_2,\cdots,a_n)\in\Lambda$, then $(a_n,a_1,a_2,\cdots,a_{n-1})$ is also in Λ.

Pairwise lattice. A lattice Λ^* generated by an arbitrary vector $v\in\Lambda$ through $\Lambda^*=\{x\in\mathbb{R}^m\ s.t.\ x^Tv\in\mathbb{R}\}$ can be called the dual lattice of Λ.

q-Modular lattice. For matrices $A\in\mathbb{R}_q^{n\times m}$, vectors $u\in\mathbb{R}_q^n$, and any positive integers q, m, there are the following definitions:

$$\Lambda^{\perp}(A)=\left\{e\in\mathbb{R}^m\ s.t.\ Ae=0\left(\mathrm{mod}\,q\right)\right\} \tag{2}$$

$$\Lambda^{u}(A)=\left\{e\in\mathbb{R}^m\ s.t.\ Ae=u(\left(\mathrm{mod}\,q\right)\right\} \tag{3}$$

From this definition, it is clear that if vector $v\in\Lambda^{\perp}(A)$, then Equation $\Lambda^u(A)=\Lambda^{\perp}(A)+v$ holds. In other words, lattice $\Lambda^{\perp}(A)$ can be obtained after a displacement operation to lattice $\Lambda^u(A)$, where $\Lambda^u(A)$ is also called the companion set of $\Lambda^{\perp}(A)$.

Negligible functions. Usually denoted by $negl(n)$ and defined mathematically as given constants c and n, f is said to be negligible if there exists an integer N that satisfies $n>N$ such that the inequality $f(n)<n^{-c}$ holds.

Statistical Distance. In a finite domain S, two variables are randomly chosen to be represented by X and Y, then the statistical distance between X and Y can be calculated by the following formula:

$$\Delta(X,Y)=\frac{1}{2}\sum\nolimits_{x\in S}\left|\Pr[X=x]-\Pr[Y=x]\right| \tag{4}$$

If $\Delta(X,Y)<negl(n)$, then X and Y are statistically indistinguishable from each other. And the following inequality holds:

$$\Delta(X,Y)\le\Delta(X,Z)+\Delta(Z,Y) \tag{5}$$

2.1.2. Difficulties with the grid. The most centrally difficult problems on the lattice are the well-known Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP).The SVP refers to finding the shortest nonzero vector in lattice $L(B)$, given lattice basis B. CVP means that given a lattice basis B and a target vector t, find the closest vector in lattice $L(B)$ to t.

In order to facilitate the construction of cryptographic algorithms, some of the most basic and difficult problems on the lattice are usually transformed into certain problems of approximate difficulty by certain statutes. The most common of these are the small integer solution problem SIS and the nonchiral small integer solution problem ISIS.

Small Integer Solution Problem (SIS). Given integer q, matrix $A\in Z_q^{n\times m}$, and real number β, solve for a vector of integers $e\in Z^m$ that satisfies $Ae=0\bmod q$ and $\|e\|\le\beta$. where q,m,β can all be represented by polynomials in n, A is stochastically homogeneous, and $\|\cdot\|$ denotes the Euclidean parameter.

Nonchiral Small Integer Solution Problem (ISIS). Given integer q, matrix $A \in Z_q^{n\times m}$, and real numbers β, solve for a vector of integers $e \in Z^m$ satisfying $Ae = 0 \bmod q$ and $\|e\| \le \beta$. where $u \in Z_q^n$, q, m, β can all be represented by polynomials in n, A is randomly uniform, and $\|\cdot\|$ denotes the Euclidean parameter.

If A, u is uniformly chosen, SIS and ISIS are said to be general case hard problems. The SIS problem is equivalent to the problem of finding short nonzero vectors on a q-module lattice $\Lambda_q^{\perp}(A)$. D. Micciancio and O. Regev proved in the related literature that the SIS and ISIS problems are NP-hard with suitable choice of parameters.

2.1.3. Common algorithms on the grid:

1) Trapdoor generation algorithms. Given positive integers n, $q \ge 2$, $m > 5n\log q$. There exists a polynomial time trapdoor generation algorithm $Trap\,Gen(q,n)$ with output matrix $A \in Z_q^{n\times m}$ and a trapdoor basis $T_A \in Z_q^{n\times m}$ of lattice $\Lambda_q^{\perp}(A)$. The distribution of A is statistically indistinguishable from the uniform distribution on $Z_q^{n\times m}$ and T_A satisfies that $\|T_A\| \le O(n\log q)$, $\|\overline{T}_A\| \le O\left(\sqrt{nlbq}\right)$ holds. where $\overline{T}_A$ denotes the basis after Gram-Schmidt orthogonalization of T_A.

2) Lattice basis delegation algorithm. Given a $A \in Z_q^{n\times m}$ is a full rank matrix, T is a set of bases of lattice $\Lambda^{\perp}(A)$, Gaussian parameter σ satisfies $\sigma > \|T_A\| \cdot \sigma_R \sqrt{m} \cdot \omega\left(l^{b3/2}/m\right)$, Gaussian parameter σ_R satisfies $\sigma_R = \sqrt{nlbq} \cdot \omega(\sqrt{lbm})$, and a matrix $R \in D_{m\times m}$, where $D_{m\times m}$ denotes the distribution of matrices in $Z^{m\times m}$ that satisfy $(D_{\sigma_R}^m)^m$ and are modulo q invertible. Then there exists a probabilistic polynomial time algorithm $Basis\,Del(A,R,T,\sigma)$ capable of outputting a set of bases T_B of lattice $\Lambda^{\perp}(AR^{-1})$ such that $\|T_B\| < \sigma/\omega(\sqrt{lbmq})$.

3) Rejection sampling algorithm. For any $v \in Z^m$, there is $\sigma = \omega\left(\|v\|\sqrt{\log m}\right)$ that:

$$\Pr\left[\frac{D_{\sigma}^{\alpha}(z)}{D_{v,\alpha}^m(z)} = O(1) : z \leftarrow D_{\sigma}^m\right] = 1 - 2^{-\omega(\log m)} \tag{6}$$

Let $V = \{v \in Z^m : \|v\| < t\}$, $\sigma = \omega(t\sqrt{\log m})$. $h : V \to R$ be a probability distribution, then there exists a constant $M = O(1)$ such that the statistical distance between the output distributions of the following two algorithms is less than $2^{-\omega(\log m)}/M$.

Sample $v \leftarrow h$, $z \leftarrow D_{v,\alpha}^m$ and output (z,v) with probability $\min\left(\frac{D_{v,\alpha}^m(z)}{MD_{v,\alpha}^m(z)}, 1\right)$.

Sample $v \leftarrow h$, $z \leftarrow D_{\alpha}^m$ and output (z,v) with probability $\frac{1}{M}$.

where the probability that the first sampling of the algorithm will have an output is at least $\frac{1-2^{-\omega(\log m)}}{M}$.

4) Statistical distance for generic hash functions. Suppose there is a generic family of hash functions $\{H_z : Z_q^n \to Z_q\}_{z\in Z}$, for any random variable $W \in Z_q^n$ and any random variable Y, there are

$$SD\left((U_Z, H_Z(W), Y), (U_Z, U, Y)\right) \le \frac{1}{2}\sqrt{2^{-\bar{H}_{\infty}(W|Y)} \cdot \left|Z_q\right|} \tag{7}$$

where U_Z and U are uniformly randomly distributed over Z_q^n and Z_q. In particular, this generalized hash function is on average ε statistically close to unity.

The statistical distance SD between two probability distributions A and B is defined as:

$$SD(A,B)=\frac{1}{2}\sum v\,|\Pr(A=v)-\Pr(B=v)| \tag{8}$$

The predictability of random variable A is $\max_a \Pr[A=a]$ and the average minimum entropy $\bar{H}_\infty$ is defined as $-\log\max_a \Pr[A=a]$. The minimum entropy can be regarded as the worst-case entropy.

2.2. Chameleon Signatures and Hash Functions

2.2.1. Chameleon Signature. Let the message space be M, the random number space be R and the signature value space be S. A standard CS scheme consists of three main participants, i.e., the signer S, the designated verifier V and the arbiter [18].

1) Setup. Inputs security parameter λ and outputs public parameter pp.

2) KeyGen. Inputs pp and outputs public-private key pairs $(pk_{\mathrm{S}}, sk_{\mathrm{S}})$ and (pk_V, sk_V) for S and V.

3) Sign. Probabilistic algorithm run by S. Input pp, public key pk_V, public-private key pair $(pk_{\mathrm{S}}, sk_{\mathrm{S}})$ and message $\mu\in\mathrm{M}$, output random number $r\in\mathrm{R}$ and signature value $\sigma\in\mathrm{S}$.

4) Verify. Deterministic algorithm run by V. Input pp, public key pk_{S} and pk_V, $\mu\in\mathrm{M}$, $r\in\mathrm{R}$ and $\sigma\in\mathrm{S}$, output 1 or 0.

5) Forge. Probabilistic algorithm run by V. Input pp, public key pk_{S}, public-private key pair (pk_V, sk_V), $\mu\in\mathrm{M}$, $r\in\mathrm{R}$ and $\sigma\in\mathrm{S}$ generated by S running algorithm Sign, output new message $\mu'\in\mathrm{M}$ and random number $r'\in\mathrm{R}$ satisfying $Verify(pp, pk_{\mathrm{S}}, pk_V, \mu', r', \sigma)\to 1$.

A secure CS scheme needs to satisfy the following properties:

1) A CS scheme is said to be strongly unforgeable under an adaptive choice message attack if for any PPT the advantage $ADV_{CS,\mathrm{A}}^{Unforgeability}$ of an adversary A winning the following game is negligible [19].

2) A CS scheme is said to be nondeliverable if the advantage $ADV_{CS,\mathrm{A}}^{Non\text{-}transferability}$ for an adversary of any PPT A winning the following game is negligible.

3) A CS scheme is said to satisfy signer rejectability if $(\mu,r,\sigma)\in\mathrm{M}\times\mathrm{R}\times\mathrm{S}$ is forged by a designated verifier V and the signer S has the ability to persuade the arbiter J to reject the signature. Conversely, a CS scheme is said to satisfy signer non-repudiation if (μ,r,σ) it is authentically generated by S and it cannot be denied.

2.2.2. Chameleon Hash Function. A chameleon hash is a trapped one-way hash function where only the person with the trapped key can find the collision of the chameleon hash. A chameleon hash contains the following 5 algorithms:

$Setup_{CH}(\lambda)\to pp_{CH}$: Input security parameter λ, output public parameter pp_{CH}.

$KeyGen_{CH}(pp_{CH})\to(sk_{CH}, pk_{CH})$: Input public parameter pp_{CH} and output key pair (sk_{CH}, pk_{CH}), where sk_{CH} is a trapdoor.

$Hash_{CH}(m, pk_{CH})\to(h,r)$: Input message m and hashed public key pk_{CH}, output chameleon hash h and corresponding chameleon random number r.

$Verify_{CH}(m,h,r,pk_{CH})\to(0,1)$: Input message m, hash value, chameleon random number r, and public key pk_{CH}, and output verification result. If the verification passes, output 1, otherwise, output 0.

$Adapt_{CH}(m,h,r,m',sk_{CH})\to r'$: Input message m, hash value, chameleon random number r, new message m' and trapdoor sk_{CH}, output new chameleon random number r' [20].

3. Gegami's identity-based chameleon signature scheme

The rapid development of Internet technology has led the society to be in the environment of information data explosion, and the efficiency and convenience of the Internet have greatly improved people's quality of life, but at the same time we are also facing information security hazards such as malicious data and privacy exposure. Therefore, people have begun to study how to avoid the exposure of personal information, resist attacks by malicious users, and ensure the mechanism of reliable information sources on the network. Lattice ciphers, which have the property of being as hard in the average case as in the worst case, are considered to be an important tool for resisting attacks on quantum computers and have been widely studied. Moreover, lattice cipher algorithms mainly use integer lattices, and the general operation objects are matrices and vectors, which basically do not involve complex algorithms such as exponential operations or bilinear pairs, so the computational speed of lattice cipher system has a significant advantage over the traditional public key cryptography system.

3.1. Identity-based chameleon hashing on grids

3.1.1. Identity-based chameleon hashing. The identity-based chameleon hash function consists of the following 4 algorithms:

1) Setup: a probabilistic polynomial time algorithm controlled by security parameter k that generates a key pair (K_S, K_P). A system parameter P_{SYS} containing K_P the secret master key K_S is published by PKG.

2) Extract: a deterministic polynomial-time algorithm with inputs the master key K_S and identity string I and output I the corresponding trapdoor key K_T .3) Hash: a probabilistic polynomial-time algorithm with inputs 1 identity string I, 1 message m, and 1 random string r; the output is 1 hash value $h = Hash(I, m, r)$. Here h is independent of K_T.

4) Forge: a deterministic polynomial time algorithm F where the input is a trapdoor key K_T corresponding to an identity string I, for 1 hash value h of message m, 1 random string r and another message $m' \neq m$, and the output is 1 string r' satisfying the following relation:

$$h = Hash(I, m, r) = Hash(I, m', r') \tag{9}$$

The security of identity-based chameleon hash functions consists of two aspects:

1) collision-resistant forgery under active attack ordering ID to be the target identity and m to be the target message, for any 4 polynomial f_1, f_2, f_3, f_4 (none of which are constants), and for sufficiently large k there exists no such probabilistic algorithm A which runs in less $f_1(k)$ time, performs at most $f_2(k)$ Extract interrogation, and succeeds in computing a random number r, r' and a binary string $m', m' \neq m$ with probability greater than $1/f_3(k)$ satisfying the

$$Hash(PK, ID, m, r) = Hash(PK, ID, m', r') \tag{10}$$

where the security parameter of Hash is $f_4(k)$.

2) Semantic security A Any two messages m_0, m_1 are sent to the prediction machine O, O A random uniform coin is flipped to get $b, b \in \{0,1\}$, and the Hash value of message m_b is computed h and sent to A. A It is up to m_0, m_1, h to answer whether $b = 0$ or $b = 1$, and the advantage of answering correctly is negligible. Intuitively, it is infeasible to determine from the Hash value which message it was generated from.

3.1.2. Lattice Chameleon Hash Function Construction. Based on the identity-based chameleon hash function, this paper gives the construction of the identity-based chameleon hash function on lattice and proves that it belongs to the chameleon hash function on lattice, where $G = I_w \otimes g^t \in \square_q^{w \times wk}$ is the open trapdoor matrix.

Let $w,q \in \square$ be a polynomial in the security parameters κ, $p = O(w\log_2 q)$, $k = \lceil \log_2 q \rceil$ and the invertible matrix $S \in \square_q^{w\times w}$. Let matrix $R \in \square^{(p-wk)\times wk}$ be the G-trapdoor of matrix $A \in \square_q^{w\times p}$ labeled S, $s \geq s_1(R)\cdot\omega(\sqrt{\log_2 w})$. Let function $f:\{0,1\}^* \to \square_q^w$ be a collision resistant hash function. Define function CH_A to be $CH_A(m,r) = \left(f(m) + Ar\right) \bmod q$, where the independent variables $m \in \{0,1\}^*$, $r \sim D_{\square^p,s}$.

Function $CH_A(m,r)$ is a chameleon hash function based on the hard problem $ISIS_{q,p,2s\sqrt{p}}$ on the lattice.

Proof:

To prove that function $CH_A(m,r)$ is a chameleon hash function, it is only necessary to prove that function $CH_A(m,r)$ satisfies three properties of chameleon hash functions, which are collision resistance, trapdoor collision, and semantic security.

1) Collision resistance. Let finding a pair of collisions from function $CH_A(m,r)$ be no less difficult than solving the ISIS problem on the lattice. Suppose the pair of collisions in function $CH_A(m,r)$ is (m_1,r_1) and (m_2,r_2) (where $m_1 \neq m_2$), i.e., $CH_A(m_1,r_1) = CH_A(m_2,r_2)$. Is thus obtained:

$$f(m_1) - f(m_2) + A(r_1 - r_2) = 0 \bmod q \tag{11}$$

That is, $A\left(r_1 - r_2\right) = f\left(m_2\right) - f\left(m_1\right) \bmod q$. Let $z = f(m_2) - f(m_1)$, $v = r_1 - r_2$, yield $r_1 \neq r_2$, so $v \neq 0$. Substituting these variables into the above equation yields $Av = z \bmod q$. And because:

$$\|v\| = \|r_1 - r_2\| \leq \|r_1\| + \|r_2\| \leq 2s\sqrt{p} \tag{12}$$

Therefore v is a valid solution to the instance (A,z) given in $ISIS_{q,p,2s\sqrt{p}}$.

2) Trapdoor collisionality. For $r_1 \sim D_{\square^p,s}$, $m_1,m_2 \in \{0,1\}^*$ and satisfying $m_1 \neq m_2$, there is:

$$CH_A(m_1,r_1) = \left(f(m_1) + Ar_1\right) \bmod q \tag{13}$$

Let $u = (f(m_1) + Ar_1) - f(m_2)$, run the efficient algorithm $SampleD\left(R,A,S,u,s\right)$, the output vector $r_2 \sim D_{\Lambda_u^\perp(A),s}$, and thus have $Ar_2 = u \mod q$. In summary, an efficient algorithm TrapCol can be defined with input (A,R,m_1,r_1,m_2) and output r_2 satisfying $CH_A(m_1,r_1) = CH_A(m_2,r_2)$.

3) Semantic security. Taking any $m_i \in \{0,1\}^*$, $r_i \sim D_{\square^p,s}(i=1,2)$, we have:

$$CH_A(m_i,r_i) = (f(m_i) + Ar_i) \bmod q \tag{14}$$

It can be obtained that the distribution of $Ar_i \bmod q$ is approximately obeying the uniform distribution on $\square_q^w$. Also from the randomness of the selection of message m_i, it follows that the distribution of hash value $CH_A(m_i,r_i)$ is approximately obeying a uniform distribution on $\square_q^w$. Therefore, it can be obtained: hash value $CH_A(m_1,r_1)$ and $CH_A(m_2,r_2)$ of the probability distribution is computationally indistinguishable.

In summary, it can be obtained: function $CH_A(m,r)$ is a chameleon hash function on the lattice, where A is the hash key hk, R is the trapdoor key tk.

3.2. Design of Gekko's identity-based CS program

3.2.1. Identity-based CS programs. Here, we need to assume that S is the signer and R is the receiver.

Setup: the system generation algorithm, in which the private key generation organization PKG is responsible for generating the system master key and public parameters as follows: the private key

generation organization selects an additive group G on an elliptic curve, G of order prime q, $q \geq 2^k$, where k is a secure parameter, and the generating element of G is g. Then it selects a multiplicative group G_1 on the elliptic curve, and the order of G_1 is q as well. define e: $G \times G \to G_1$ is a bilinear mapping. Here we have to define four hash functions: H_0, H_1, H_2, H_3 they are denoted as:

$$H_0 : \{0,1\}^* \to G^* \tag{15}$$

$$H_1 : \{0,1\}^* \to Z_q^* \tag{16}$$

$$H_2 : G_1 \to Z_q^* \tag{17}$$

$$H_3 : G \to Z_q^* \tag{18}$$

PKG randomly selects $x \in_R Z_q^*$, where x is the master key, R satisfies equation $R = kg \in G$, and the system public key Y and $Y = xg$ can be calculated from x and g. Finally, the system public parameters can be expressed as $\{G, G_1, q, g, e, H_0, H_1, H_2, H_3, Y\}$, where the master key x is kept secret and the public parameters can be made public.

Extract: user private key generation algorithm, in order to generate the private key, for user U, he first needs to submit his identity ID_U to the private key generation organization, at this time, the public key PK_U of user U can be derived, which is denoted as $PK_U = H_0(ID_U)$, and the private key generation organization is responsible for verifying the identity of the user, and if the identity matches, the corresponding private key $SK_U = xPK_U$ will be transmitted to the user through a secure channel.

Signing: signing algorithm, this algorithm realizes the signature of the message m, the signer chooses a random number $r \in_R G$, $k \in_R Z_q^*$, and calculates the chameleon hash h of the message m, where:

$$h = H\left(Y, ID_R, m, r\right) = e(r, g)e\left(H_1(m)PK_R, Y\right) \in G_1 \tag{19}$$

$$R = kg \in G \tag{20}$$

$$S = k^{-1}\left(H_2(h)g + H_3(R)SK_V\right) \tag{21}$$

The signature of message m can be represented as (m, R, S, r).

Verification: the verification algorithm, R after receiving the signature, performs the computation on $e(R,S)$, i.e:

$$\begin{aligned} e\left(R,S\right) &= e(kg, k^{-1}H_2(h)g) + H_3(R)SK_V \\ &= e(kg, k^{-1}H_2(h)g)e(kg, k^{-1}H_3(R)SK_V) \\ &= e(g, H_2(h)g)e(g, H_3(R)SK_V) \\ &= e(g,g)^{H_2(h)}e(g, SK_V)^{H_3(R)} \\ &= e(g,g)^{H_2(h)}e(g, xPK_V)^{H_3(R)} \\ &= e(g,g)^{H_2(h)}e(xg, PK_V)^{H_3(R)} \\ &= e(g,g)^{H_2(h)}e(Y, PK_V)^{H_3(R)} \end{aligned} \tag{22}$$

yields $e(R,S) = e(g,g)^{H_2(h)}e\left(g, SK_V\right)^{H_3(R)}$, then (m, R, S, r) is determined to be the signer's legitimate signature for message m; otherwise, this signature is rejected.

3.2.2. Gegami's identity-based CS program. Let the message space $\mathrm{M} = \{0,1\}^m$ (arbitrary messages can be mapped to M using a collision-resistant hash function), the identity space $\mathrm{ID} = \{0,1\}^*$, the

random number space $\mathrm{R}=D_{Z^{m\times s}}$, and the signature space $\mathrm{S}=D_{Z^{m\times s}}$, $id_S\in \mathrm{ID}$, and $id_V\in \mathrm{ID}$ correspond to the identity of the signer and the specified verifier, respectively.

$Setup\left(1^{\lambda}\right)\to\left(pp,msk\right)$: Input security parameters into, let prime number $q=poly(\lambda)$, integer $m=2n\lceil\log_2 q\rceil$, $n=poly(\lambda)$, Gaussian parameter $s=\tilde{\mathrm{O}}\left(n^2\right)$. The PKG performs the following operations:

1) Run $TrapGen\left(q,n,m\right)$ to generate trapdoor T_A for $A\in Z_q^{n\times m}$ and $\Lambda_q^{\perp}(A)$.

2) Randomly pick $B\in Z_q^{n\times m}$ as the common matrix for constructing the chameleon hash function CH.

3) Pick collision-resistant hash functions $\mathrm{H}_1:\{0,1\}^*\to \mathrm{D}_{m\times m}$ and $\mathrm{H}_2:\{0,1\}^*\to Z_q^n$.

4) Output public parameter $pp=\left(A,B,\mathrm{H}_1,\mathrm{H}_2\right)$ and system master private key $msk=T_A$.

$KeyGen\left(pp,msk,id\right)\to\left(sk_{id}\right)$: Entering the parameters pp, the master private key msk, and the identity $id\in \mathrm{ID}$, PKG performs the following operations:

1) Order $R_{id}=\mathrm{H}_1(id)\in \mathrm{D}_{m\times m}$ and compute $F_{id}=A\cdot R_{id}^{-1}\bmod q\in Z_q^{n\times m}$ (hereafter F_{id_S} and F_{id_V} are abbreviated as F_S and F_V, respectively).

2) Run $BasisDel\left(A,R_{id},T_A,s\right)$ to generate the trapdoor $T_{F_{id}}$ of $\Lambda_q^{\perp}\left(F_{id}\right)$.

3) Output the private key $sk_{id}=T_{F_{id}}$.

$Sign\left(pp,id_S,id_V,sk_{id_S},\mu\right)\to(\mu,r,\sigma)$: Input parameters pp, signer identity $id_S\in \mathrm{ID}$, specified verifier identity $id_V\in \mathrm{ID}$, signer private key $sk_{id_S}=T_{F_S}$ and message $\mu\in\{0,1\}^m$, the signer performs the following actions:

1) First find out whether the local signature repository stores $\left(id_V,\mu,r,\sigma\right)$, if it exists, perform step (6), otherwise perform step (2).

2) Make $R_{id_S}=\mathrm{H}_1\left(id_S\right)$ and $R_{id_V}=\mathrm{H}_1\left(id_V\right)$, and compute $F_S=A\cdot R_{id_S}^{-1}\bmod q$ and $F_V=A\cdot R_{id_V}^{-1}\bmod q$.

3) Pick $r\in \mathrm{R}$ randomly and compute chameleon hash $y=\mathrm{CH}(\mu,r)=B\cdot\mu+F_V\cdot r\bmod q$ about μ.

4) Order $h=\mathrm{H}_2\left(id_S,id_V,bin(y)\right)$, where $bin\left(y\right)\in\{0,1\}^{n\lceil\log_2 q\rceil}$ is the binary representation of $y\in Z_q^n$.

5) Run Sample Pre $\left(F_S,T_{F_S},h,s\right)$ to generate the signature value $\sigma\in Z^m$.

6) Output (μ,r,σ) and store $\left(id_V,\mu,r,\sigma\right)$ in the local signature library.

$Verify\left(pp,id_S,id_V,\mu,r,\sigma\right)\to\left(1\ or\ 0\right)$: Input parameters pp, Signer Identity $id_S\in \mathrm{ID}$, Specify Verifier Identity $id_V\in \mathrm{ID}$, Message $\mu\in\{0,1\}^m$, Random Number $r\in \mathrm{R}$, and Signature Value $\sigma\in \mathrm{S}$, and Specify Verifier to perform the following actions:

1) Verify that $0<\|r\|$, $\|\sigma\|\le s\sqrt{m}$ hold.

2) Make $R_{id_S}=\mathrm{H}_1\left(id_S\right)$ and $R_{id_V}=\mathrm{H}_1\left(id_V\right)$, and compute $F_S=A\cdot R_{id_S}^{-1}\bmod q$ and $F_V=A\cdot R_{id_V}^{-1}\bmod q$.

3) Compute $y=\mathrm{CH}(\mu,r)=B\cdot\mu+F_V\cdot r\bmod q$.

4) Verify that $F_S\cdot\sigma=\mathrm{H}_2\left(id_S,id_V,bin(y)\right)\bmod q$ is valid.

5) If all the above conditions hold, output 1, otherwise output 0.

Forge $\left(pp,id_V,sk_{id_V},\mu,r,\sigma\right)\to\left(\mu',r',\sigma\right)$: Input parameter pp, specify the verifier identity id_V, private key $sk_{id_V}=T_{F_V}$, and $(\mu,r,\sigma)\in \mathrm{M}\times\mathrm{R}\times\mathrm{S}$ generated by the signer, and specify the verifier to perform the following operations:

1) Make $R_{id_V}=\mathrm{H}_1\left(id_V\right)$ and compute $F_V=A\cdot R_{id_V}^{-1}\bmod q$ and $y=\mathrm{CH}(\mu,r)=B\cdot\mu+F_V\cdot r\bmod q$.

2) Pick $\mu'\in\{0,1\}^m$ and $\mu'\ne\mu$ and compute $v=y-B\cdot\mu'\bmod q$.

3) Run Sample Pre $\left(F_V,T_{F_V},v,s\right)$ to generate random number $r'\in \mathrm{R}$.

4) Output (μ', r', σ).

4. Gerson's identity-based CS program validation analysis

The rapid development of computer technology and communication technology has accelerated the speed of social informatization, and the speed of data interaction and the scale of communication have seen unprecedented growth. However, while the high-speed development of information brings great convenience to social life, it also poses a great threat to users' privacy rights and interests. Digital signature is a technology widely used in the field of privacy protection, and it is also a hot topic in cryptography research, playing an important role in today's informationization era. However, the rapid development of quantum computing has threatened the security of digital signature schemes based on number theory regimes, and constructing cryptographic schemes that can resist quantum attacks has become an urgent requirement in today's era. Currently, no polynomial-time solution has been found for some difficult problems on the lattice, so the cryptographic schemes constructed based on the difficult assumptions on the lattice are considered to be resistant to quantum attacks.

4.1. Security analysis

4.1.1. Proof of non-falsifiability. For the security of the identity-based chameleon signature scheme on the lattice proposed in this paper, this paper proves verification in terms of unforgeability.

As long as the SIS problem is intractable, the identity-based CS scheme on the lattice designed in this paper possesses the property of unforgeability under selective message and identity attacks under the stochastic predicate machine model. The proof is as follows:

We assume that there exists an attacker A that has a running time of at most t, can perform at most q_e Extraction queries, q_h Chameleon Hash queries, and q_s Signing queries, and then is able to forge a valid signature with a non-negligible probability e, then there exists a challenger C that is able to solve the SIS problem with another non-negligible negligible probability e' of being able to solve the SIS problem. We analyze the problem as follows by looking at both A and C.

First, A declares the identity ID' of the target to be attacked, and forges a signature with the identity ID'' of the receiver (which remains unchanged during the process).

Setup: C Perform the following operation. Choose two matrices $A, B \leftarrow \square^{n\times m}$ uniformly at random. Run $Trap\ Gen(q,n,m)$ to generate a matrix pair $(A_0, T_0) \in \square^{n\times m} \times \square^{m\times m}$, where T_0 is a set of short bases of lattice $\Lambda_q^{\perp}(A_0)$, we make the master public key A_0 and the master private key T_0, and disclose all information other than the master private key T_0 to A.

Extraction queries: A can adaptively perform Extraction queries on the identity $ID(ID \neq ID')$, C creating a list $L_1 = (ID, SK_{ID})$ that is initially the empty set.

C uses the $ExtBasis(\cdot)$ algorithm to find the identity corresponding to the private key: $SK_{ID} \leftarrow ExtBasis(T_0, A_0 \| A + H(ID_U)B)$, and therefore SK_{ID} is also a set of short bases for $\Lambda_q^{\perp}(A_0 \| A + H(ID_U)B)$. C returns SK_{ID} to A, which in turn deposits (ID, SK_{ID}) into list L_1.

Chameleon hash queries: A perform a query on the corresponding hash value of any message μ. C creates and maintains a list $L_2 = (\mu, y)$ that is initially the empty set.

C First find out if there is a hash value y from list L_2, if so, return it to A. If not, select $\sigma \leftarrow Sample\ Dom(1^{2m})$ and compute $y = f_{F_{ID}}(\sigma)$ and return it to A, and store (μ, y) in list L_2.

Signing queries: A choose any identity ID, a message μ, to C initiate a query. C create a list $L_3 = (ID, \mu, y, \sigma)$ with an empty set initially.

C First look in L_2 to see if there is a hash value y for message μ, if so use it, if not run Chameleon hash queries to compute it. Then C looks in list L_1 to see if there is a private key SK_{ID} that corresponds to identity ID. If the private key has been saved in the list, then it can be used directly to sign SK_{ID}. If it does not exist, we first perform Extraction queries to compute a corresponding private key SK_{ID}, and then use the Sample Pre algorithm to generate a signature $\sigma \leftarrow Sample\Pr e\left(F_{ID}, SK_{ID}, y, s_2\right)$, then C returns the signature σ to A, and then $\left(ID, \mu, y, \sigma\right)$ is stored in list L_3.

Forge: select the receiver with identity ID'', select identity ID' and message μ', A compute the next legitimate signature σ' for message μ' identity ID', and output $\left(ID', \mu', \sigma'\right)$. According to the game we have designed, A the identity ID' is never queried during the Extraction queries phase, so we consider A that we have successfully forged a valid signature as long as $\left(ID', \mu', \sigma'\right)$ the following two conditions are met that obtains the game:

1) Verify $\left(ID', \mu', r', \sigma'\right) = 1$;

2) A never made a query about $\left(ID', \mu'\right)$ the relative signature σ' during the Signingqueries phase.

Before A forging the signature, the corresponding hash y' of message μ' has been queried, and when A outputting the signature, C executing the signature algorithm outputs a quaternion $\left(ID', \mu', f'_{F_{ID}}\left(\sigma^*\right), \sigma^*\right)$, and $\sigma^* \neq \sigma'$. Then a pair of collisions is found at this time $\left(\sigma^*, \sigma'\right)$. Since the outputs of the Sample Dom $\left(1^{2m}\right)$ algorithm and the outputs of the chameleon hash function are uniformly random and have the same domain of values, at A the time of outputting a legal signature $\left(ID', \mu', \sigma'\right)$ time, we have $f'_{F_{ID}}\left(\sigma^*\right) = y' = f'_{F_{ID}}\left(\sigma'\right)$, then $y' \cdot \sigma^* = y' \cdot \sigma'$, which is $y' \cdot \left(\sigma^* - \sigma'\right) = 0$. We make $B = y'$. since $\sigma^* \neq \sigma'$, one can view $\left(\sigma^* - \sigma'\right)$ as a solution to $Bx = 0$, which is equivalent to solving the SIS problem.

4.1.2. Resistance to quantum attacks. In this paper, an identity-based CS scheme on the lattice is proposed, which aims to further enhance the resistance to quantum attacks, based on which simulation verification analysis is carried out in this paper. By constantly adjusting the ratio of malicious nodes and malicious data in the simulation experiments, we compare the traditional CS scheme (A), the identity-based CS scheme (B) and the identity-based CS scheme on the lattice (C) in this paper, and test the ability of these three methods to resist quantum computer attacks. Where the resistance to quantum attacks includes increasing the processing time for malicious nodes with the number of malicious nodes and increasing the correct modification rate with the proportion of malicious data.

The simulation experiment simulates a federated blockchain with 120 nodes, of which there are 20 signature nodes and 20 verification nodes each, and sets up different numbers of malicious node data to test the impact of the ability to resist quantum attacks. The processing time, i.e., the time required to remove the malicious nodes, and the impact of different numbers of malicious nodes on the processing time is shown in Fig. 1.

As can be seen from the figure, with the increase in the number of malicious nodes, the processing time of the traditional CS scheme and the identity-based CS scheme is gradually rising, while the identity-based CS scheme on the grid is completed by the signature nodes and verification nodes in collaboration with the modification of the federated blockchain data, only by attacking the selected signature nodes and verification nodes, and the signature and modification process requires the consent of greater than 50% of identity nodes, in order to complete the modification. Therefore, with the increasing number of malicious nodes, when the number of malicious nodes reaches 50, the

processing time is still guaranteed to be fast (6.44s). As the number of malicious nodes increases, the processing time of this paper's method improves by 50.27% and 34.68% over the processing efficiency of the traditional CS scheme and the identity-based CS scheme, respectively. The experimental results show that the identity-based CS scheme on the grid designed in this paper has high security.

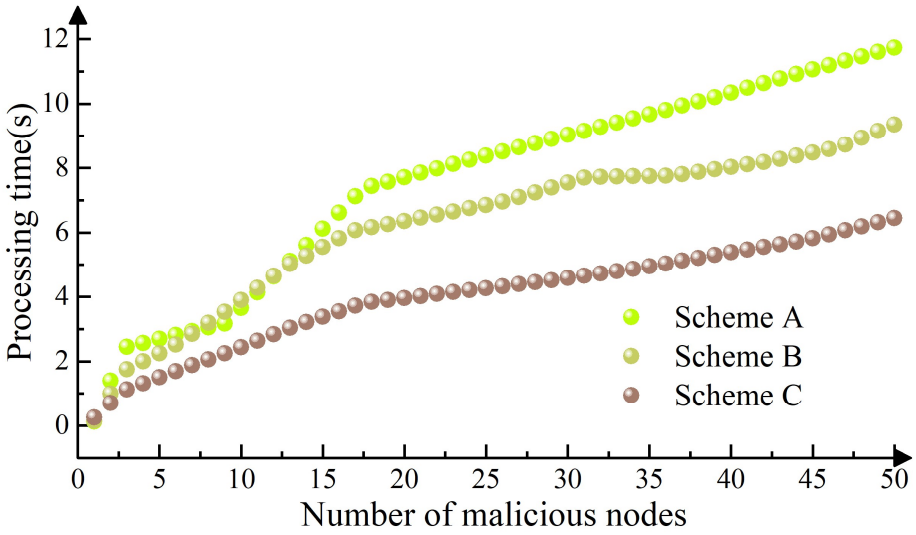


Fig. 1. Different numbers of malicious nodes on processing time

Correct modification rate i.e. the percentage of data that will be modified correctly by the node every hour, the effect of different percentage of malicious data on the correct modification rate is shown in Fig. 2. As can be seen from the figure, both the traditional CS scheme and the identity-based CS scheme are less efficient in correctly modifying the malicious data as the proportion of malicious data number increases. After processing more than 12 malicious data at the same time, this paper's method has a significant advantage over the other two methods. Due to the authentication mechanism proposed in this paper's method, it improves the node's correct modification rate for malicious data, and still maintains a high correct modification rate as the proportion of malicious data increases. When the proportion of malicious data reaches 50% (i.e., 50 pieces of data), this paper's method improves the correct modification rate by 47.48% and 42.69% compared to the traditional CS scheme and the identity-based CS scheme, respectively. The experimental results show that the method in this paper has high security.

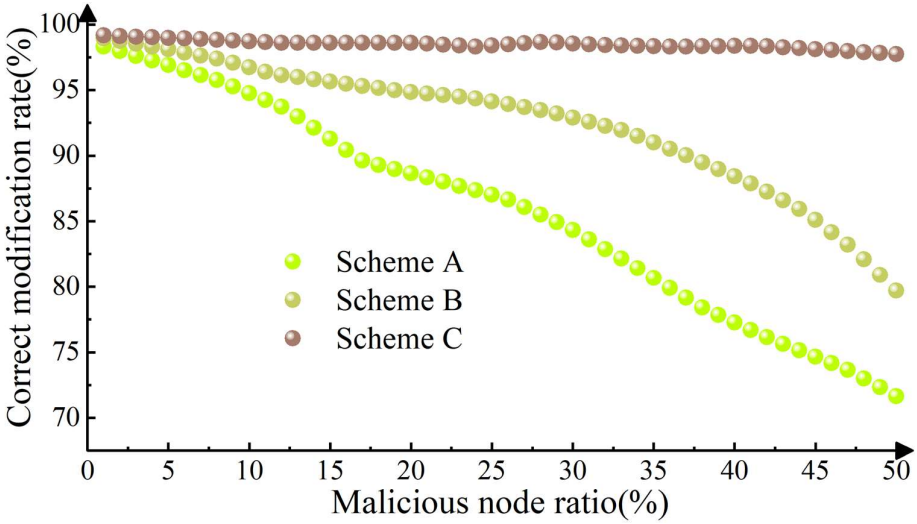


Fig. 2. Different proportions of malicious data on correct modification rate

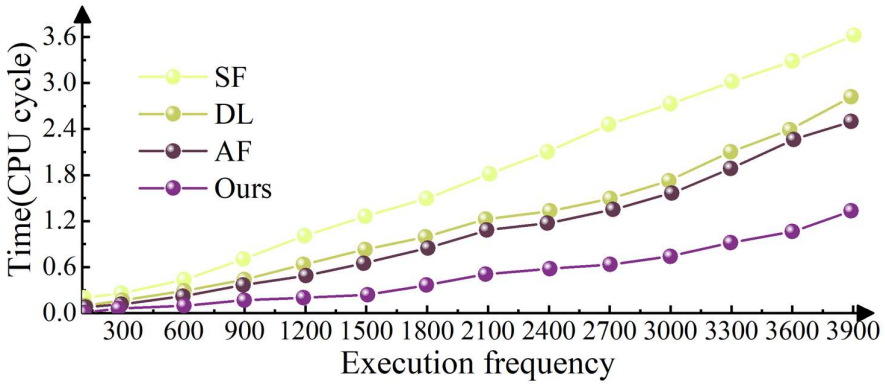
In summary, as the proportion of malicious attacking nodes and data in quantum computers increases, the identity-based CS scheme on the lattice proposed in this paper still maintains a good correct modification rate and processing time. The identity verification mechanism of this paper's method discovers and corrects the malicious attacking nodes through the identity nodes, which can improve the motivation of the nodes to check and correct the malicious attacks in time, and improve the overall data security of the system.

4.2. Chameleon Hash Verification

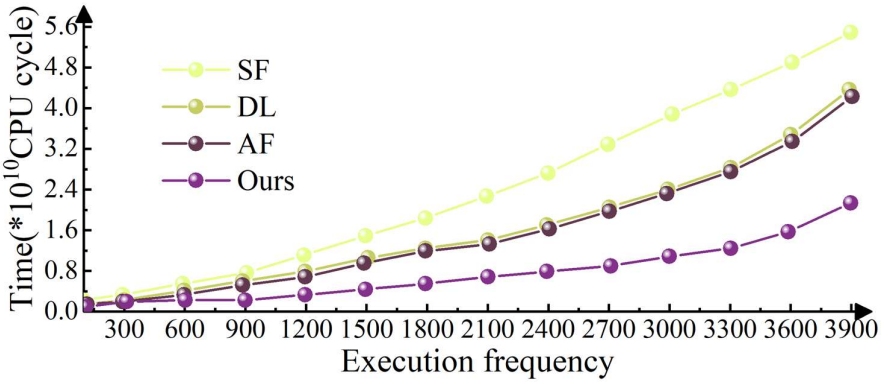
4.2.1. Comparison of Chameleon Hash Functions. For the chameleon signature scheme, the construction method of chameleon hash function plays a crucial role in improving its efficiency. This paper proposes a lattice-based chameleon hash function construction method based on the formal definition of identity, and in order to verify the effectiveness of the method, this paper selects the simple factorization (SF) based, discrete logarithm (DL) based, and advanced factorization (AF) based chameleon hash function construction methods as comparisons to repeatedly execute the Signing and Verification algorithms' time consumption and analyze their performance differences as a way to verify the feasibility of identity-based chameleon hash function construction on this paper's grid.

This experiment is carried out on VM ware Workstation Pro virtual machine, the physical machine is Windows Professional 64-bit operating system, the processor is Intel(R) Core(TM) i3-10100 CPU @ 3.60GHz 3.60 GHz, the system memory 16.0 GB, the virtual machine system is Ubuntu, the system memory 2 GB. In order to get a more accurate running time consumption of the algorithm, this experiment takes the CPU cycles during the execution of the algorithm as the running time of the algorithm. The Signing and Verification algorithms are called n times respectively in the experiment, and the statistics of time consumption under different types of chameleon hash function implementations are shown in Fig. 3. Where Figure 3(a)~(b) shows the time consumption of Signing and Verification algorithms respectively.

It can be seen that the SF-based chameleon hash function obviously consumes more time in executing the Signing and Verification algorithms when the algorithms are invoked the same number of times respectively, while the DL- and AF-based chameleon hash functions consume slightly less time in AF although they are similar. The identity-based chameleon hash function on the lattice proposed in this paper consumes the lowest time in executing the Signing and Verification algorithms, and its maximum time consumed in executing the two algorithms is 1.33*1010 CPU cycles and 2.14*1010 CPU cycles, respectively. This trend becomes more obvious as the number of executions increases, so the CS scheme execution has the best time performance when using the chameleon hash function implemented in the Gegami way in the Gegami identity-based CS scheme.



(a) Signing



(b) Verification

Fig. 3. Execution time consumption

4.2.2. Data update time. After verifying the feasibility of the construction of the identity-based chameleon hash function on the lattice, the purpose of applying this hash function in the CS scheme is to enhance the security of digital signatures, which can have a better execution time regardless of the signature length. As a result, this paper explores the data update time of the identity-based chameleon hash function on the lattice under different signature lengths in simulation experiments as a way to check the operation efficiency of the hash function. Figure 4 shows the data update time of the chameleon hash.

From the data distribution in the figure, it can be seen that with the increase in the number of data updates, when the signature character length of 60, 120, 240 characters when the data update time does not differ much, especially in the number of updates for 200 times or less, the time required for its operation is about 1ms. And as the number of data updates increases, the update time increases significantly each time the signature character length is greater than 120 characters, especially when the number of updates reaches 960, the algorithm data update time difference is large. When the signature character length is shorter, the time required for data update is shorter and can support multiple data updates within a short period of time, but when the signature character length is longer, the time impact caused by the change in the number of data updates is larger. For each fixed signature character length, the total update time and total time increase significantly as the number of updates increases. For each fixed number of updates, an increase in signature character length also results in an increase in total time. Thus, the effect of the number of updates on the total time is linear, while the effect of the signature character length on the total time is exponential.

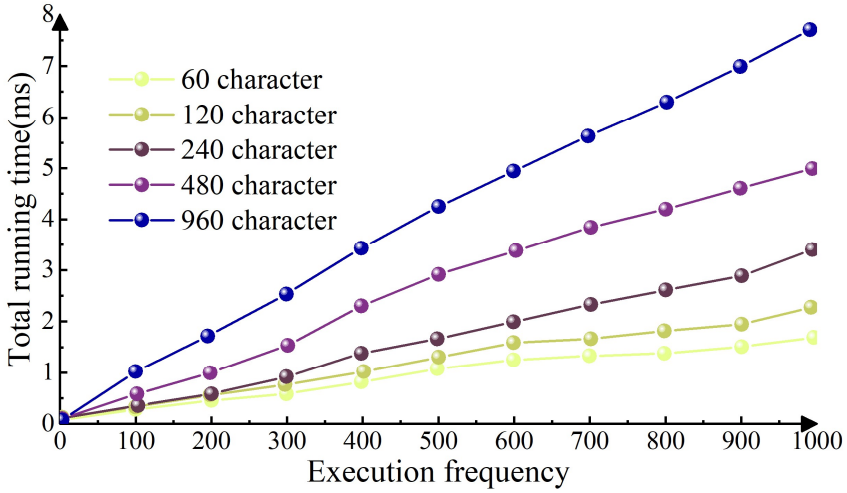


Fig. 4. Change the time of the chameleon hash

4.3. Program efficiency analysis

4.3.1. Comparison of program efficiency. In this paper, we propose an identity-based CS scheme on the lattice, which uses the stochastic predicate machine model as a security model, and in order to verify the efficiency of the scheme, it is analyzed in comparison with other signature schemes resistant to quantum attacks in terms of functionality, storage and transmission costs. Table 1 shows the efficiency analysis results of different schemes, where k_0 and k_1 denote the dataset size of homomorphic computation and the number of internal fixed points of directed acyclic graphs, respectively, and ×, √, and - denote unsatisfiable, satisfiable, and disregarded, respectively. pu, si, un, in, de, re, and lo denote the length of the public parameter, the length of the signature, the unforgeability, the nondeleterability, the rejection ness, non-repudiation, and no local storage required.

From the functional aspect, the identity-based CS scheme on lattice proposed herein combines the strong security assurance of the lattice cryptosystem against quantum computing attacks and the provable security scheme under the randomized predicate machine model, avoids the security loophole that any third party can forge the signature and the signer can not reject the signature forged by the designated verifier in Scheme A, solves the controversial issue with respect to the signature in Scheme B, and makes up for the security loophole that any third party can forge the signature and the signer can not reject the signature forged by the designated verifier in Scheme E. The gap of provably secure lattice-based CS under the stochastic predicate machine model.

In terms of storage and transmission cost, the identity-based CS on lattice scheme proposed in this paper is similar to the provably secure identity-based CS on lattice under the stochastic predicate machine model given in Scheme E. The random matrix in the final signature is used as a public parameter of the system, and it is no longer sent together with the random number $r \in R$ and the signature value $\sigma \in S$ after each selection by S, which reduces the transmission cost of the signature in Scheme A. The public parameters only contain a random matrix on $Z_q^{n\times m}$, $n+1$ random vectors on Z_q^n and a collision-resistant hash function with an output of a n long bit string, and the final signature only contains a message $\mu \in M$ and two short vectors on $D_{Z^m,S}$, which reduces the storage cost of the public parameters in Schemes A~E and improves the transmission efficiency of the signature. And the signature process adopts a random coding strategy based on the assumption of the LWE puzzle, with public key encryption of message $D_{Z^m,S}$ and random coding of random numbers $r \in R$, one chameleon hash computation and one Gaussian proto-image sampling, and the final signature contains only two short vectors on one message $\mu \in M, D_{Z^m,S}$ and three random vectors on Z_q^m, i.e., the length of the signature is increased, but the asymptotic complexity is still the same as that in Scheme E. In particular, the signer's dependence on the local signature repository in Schemes A~E is completely eliminated, resulting in greater savings in local storage costs.

Table 1. Analysis of the efficiency of different schemes

Scheme	PU	SI	UN	IN	DE	RE	LO
A	$\tilde{O}(n^2)$	$\tilde{O}(n^2)$	×	√	×	√	×
B	$\tilde{O}(n^3)$	$\tilde{O}(n^2)$	√	√	×	√	×
C	$\tilde{O}(k_0 \cdot n^2)$	$\tilde{O}(n^2)$	√	×	—	—	—
D	$\tilde{O}(n^2)$	$\tilde{O}(k_1 \cdot n)$	√	√	√	√	×
E	$\tilde{O}(n^2)$	$\tilde{O}(n)$	√	√	√	√	×
Ours	$\tilde{O}(n^2)$	$\tilde{O}(n)$	√	√	√	√	√

4.3.2. Storage computation overhead. For the performance of the identity-based CS scheme on lattice proposed in this paper, it is mainly verified in two dimensions: storage overhead and computation overhead, and the attribute-based signature scheme on lattice (O1) and the identity-based CS scheme (O2) are selected for comparison. The storage overhead of the signature scheme

mainly considers the key length and the signature length, and the public key length, the private key length and the generated signature length data of the signer are compared in the comparison process. This comparison experiment is carried out in 100bit security level, following which the storage and computation overheads of different schemes are obtained as shown in Table 2.

From the data in the table, it can be seen that the identity-based CS scheme on lattice designed in this paper has significant advantages in terms of public key length and signature length overhead. The lattice-based attribute-based signature scheme not only adopts the bonsai tree technique to realize the key expansion, but also adopts the signature method in the Cash scheme, so the storage overhead is relatively high. The identity-based CS scheme, on the other hand, proposes an identity-based CS technique, which adopts a reassignment method for the matrix in the scheme to avoid direct increment on the original matrix. Therefore, scheme O2 reduces the public key and signature length compared to scheme O1. In this paper, the proposed scheme is based on the identity-based CS scheme, combined with the construction of the Gerber chameleon hash function, followed by the establishment of the Gerber identity-based CS scheme used in the generation of the key. After that, in the signing step, the signature method in the GPV scheme is adopted to complete the message processing step, which makes the scheme in this paper further reduce the length of the public key and signature compared with the previous schemes. In addition, in the verification process of computational overhead, since the generated matrix parameters are the same and the same sampling technique is used, the difference in the verification overhead of the three schemes is only 2.05%, and the difference in the signature overhead is only 0.73%. Therefore, it is fully demonstrated that the identity-based CS scheme on the lattice proposed in this paper can maintain a low computational overhead on the basis of reducing the storage overhead, which supports the improvement of the signature scheme's performance against quantum attacks.

Table 2. Storage and computing cost of different schemes

	Index	Scheme O1	Scheme O2	Ours
Storage costs	Public key length	18.25MB	19.46MB	9.37MB
	Private key length	124.69MB	246.72MB	1327.51MB
	Signature length	36.84KB	20.35KB	16.42KB
Compute cost	Signature cost	2.71*104ms	2.72*104ms	2.69*104ms
	Signature cost	1.45ms	1.46ms	1.43ms

5. Conclusions and outlook

5.1. Conclusion

Aiming at the shortcomings of the traditional CS scheme that cannot resist quantum computer attacks, this paper introduces the lattice cipher to establish the identity-based CS scheme on the lattice, and carries out the verification and analysis of its security, quantum attack resistance, efficiency, and storage and computation overhead. The scheme designed in this paper can ensure the processing time within 6.5s even if the number of malicious nodes reaches 50, and has the advantages of non-forgeability, non-transmissibility, deniability, non-repudiation, and no need for local storage. Gegami's identity-based chameleon hash function construction method can significantly improve its data update time and provide assistance to enhance the computational efficiency of the CS scheme. The identity-based CS scheme on lattice designed in this paper has significant resistance to quantum attacks and provides a new research direction to ensure the security of digital signatures and information transmission.

5.2. Outlook

According to the research results of this paper, the future optimization of the identity-based chameleon signature scheme on the lattice can be carried out in the following two aspects:

First, an efficient multi-key threshold fully homomorphic encryption scheme can be considered to be designed to utilize additive homomorphism and multiplicative homomorphism, which makes it more flexible and can cope with a wider range of application requirements.

Second, one can consider proving the security of the scheme under the standard model to further improve the reliability and practicality of the scheme.

References

[1] Alagheband, M. R., & Mashatan, A. (2022). Advanced digital signatures for preserving privacy and trust management in hierarchical heterogeneous IoT: Taxonomy, capabilities, and objectives. Internet of Things, 18, 100492.

[2] Shankar, G., Ai-Farhani, L. H., Anitha Christy Angelin, P., Singh, P., Alqahtani, A., Singh, A., ... & Samori, I. A. (2023). Improved multisignature scheme for authenticity of digital document in digital forensics using edward-curve digital signature algorithm. Security and Communication Networks, 2023(1), 2093407.

[3] Li, Q., He, D., Chen, Y., Wen, J., & Yang, Z. (2024). An efficient quantum-resistant undeniable signature protocol for the E-voting system. Journal of information security and applications, 81, 103714.

[4] Rastegari, P., Berenjkoub, M., Dakhilalian, M., & Susilo, W. (2019). Universal designated verifier signature scheme with non-delegatability in the standard model. Information Sciences, 479, 321-334.

[5] Thanalakshmi, P., Anitha, R., Anbazhagan, N., Cho, W., Joshi, G. P., & Yang, E. (2021). A hash-based quantum-resistant chameleon signature scheme. Sensors, 21(24), 8417.

[6] Xu, J., Yu, Y., Meng, Q., Wu, Q., & Zhou, F. (2021). Role-based access control model for cloud storage using identity-based cryptosystem. Mobile Networks and Applications, 26, 1475-1492.

[7] Li, W., Zhang, X., Lin, S., Ban, X., & Chen, X. (2022, August). Chameleon DNN Watermarking: Dynamically Public Model Ownership Verification. In International Conference on Information Security Applications (pp. 344-356). Cham: Springer Nature Switzerland.

[8] Li, C., Jiao, X., Feng, X., Hu, A., Shen, Q., & Wu, Z. (2025). Identity-Based Chameleon Hashes in the Standard Model for Mobile Devices. IEEE Transactions on Information Forensics and Security.

[9] Sageloli, É., Pébereau, P., Méaux, P., & Chevalier, C. (2023, May). Shorter and faster identity-based signatures with tight security in the (Q) ROM from lattices. In International Conference on Applied Cryptography and Network Security (pp. 634-663). Cham: Springer Nature Switzerland.

[10] Ukwuoma, H. C., Gabriel, A. J., Thompson, A. F., & Alese, B. K. (2022). Quantum attack-resistant security system for cloud computing using lattice cryptography. International Journal for Information Security Research, 12(1), 1053-1061.

[11] Gitonga, C. K. (2025). The Impact of Quantum Computing on Cryptographic Systems: Urgency of Quantum-Resistant Algorithms and Practical Applications in Cryptography. European Journal of Information Technologies and Computer Science, 5(1), 1-10.

[12] Li, Y., & Liu, S. (2024). Identity-based chameleon hash from lattices. Journal of Computer Security, 32(6), 509-531.

[13] Li, F., Wang, J., Shang, M., Zhang, D., & Li, T. (2023). Research on Quantum-Attack-Resistant Strong Forward-Secure Signature Schemes. Entropy, 25(8), 1159.

[14] Wang, L., Huang, C., & Cheng, H. (2021, March). Quantum attack-resistant signature scheme from lattice cryptography for WFH. In 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE) (pp. 868-871). IEEE.

[15] Sabani, M. E., Savvas, I. K., Poulakis, D., Garani, G., & Makris, G. C. (2023). Evaluation and comparison of lattice-based cryptosystems for a secure quantum computing era. Electronics, 12(12), 2643.

[16] Sun, Z., Gu, C., & Zheng, Y. (2020). A review of sieve algorithms in solving the shortest lattice vector problem. IEEE Access, 8, 190475-190486.

[17] Althobaiti, O. S., & Dohler, M. (2021). Quantum-resistant cryptography for the internet of things based on location-based lattices. IEEE Access, 9, 133185-133203.

[18] Thanalakshmi P.,Anitha R.,Anbazhagan N.,Cho Woong,Joshi Gyanendra Prasad & Yang Eunmok. (2021). A Hash-Based Quantum-Resistant Chameleon Signature Scheme. Sensors,21(24),8417-8417.

[19] Thanalakshmi P. & Anitha R. (2019). A Quantum Resistant Chameleon Hashing and Signature Scheme. IETE Journal of Research,68(3),1-12.

[20] Yong Wang & Eddie Shahril Ismail. (2024). Tightly-Secure Two-Tier Signatures on Code-Based Digital Signatures with Chameleon Hash Functions. Mathematics,12(15),2375-2375.