Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057882

ARTICLE

An Efficient Anti-Quantum Blind Signature with Forward Security for Blockchain-Enabled Internet of Medical Things

Gang Xu[1,2,6], Xinyu Fan[1], Xiu-Bo Chen[2], Xin Liu[4], Zongpeng Li[5], Yanhui Mao[6,7] and Kejia Zhang[3,*]

[1]School of Information Science and Technology, North China University of Technology, Beijing, 100144, China

[2]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[3]School of Mathematical Science, Heilongjiang University, Harbin, 150080, China

[4]School of Digtial and Intelligence Industry, Inner Mongolia University of Science and Technology, Baotou, 014010, China

[5]Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, 100084, China

[6]Yunnan Key Laboratory of Blockchain Application Technology, Kunming, 650233, China

[7]Yunnan Innovation Institute of Beihang University, Kunming, 650233, China

*Corresponding Author: Kejia Zhang. Email: zhangkejia@hlju.edu.cn

Received: 29 August 2024 Accepted: 08 November 2024 Published: 17 February 2025

ABSTRACT

Blockchain-enabled Internet of Medical Things (BIoMT) has attracted significant attention from academia and healthcare organizations. However, the large amount of medical data involved in BIoMT has also raised concerns about data security and personal privacy protection. To alleviate these concerns, blind signature technology has emerged as an effective method to solve blindness and unforgeability. Unfortunately, most existing blind signature schemes suffer from the security risk of key leakage. In addition, traditional blind signature schemes are also vulnerable to quantum computing attacks. Therefore, it remains a crucial and ongoing challenge to explore the construction of key-secure, quantum-resistant blind signatures. In this paper, we introduce lattice-based forward-secure blind signature (LFSBS), a lattice-based forward-secure blind signature scheme for medical privacy preservation in BIoMT. LFSBS achieves forward security by constructing a key evolution mechanism using a binary tree structure. This mechanism ensures that even if future encryption keys are leaked, past data can still remain secure. Meanwhile, LFSBS realizes post-quantum security based on the hardness assumption of small integer solution (SIS), making it resistant to potential quantum computing attacks. In addition, we formally define and prove the security of LFSBS in a random oracle model, including blindness and forward-secure unforgeability. Comprehensive performance evaluation shows that LFSBS performs well in terms of computational overhead, with a reduction of 22%–73% compared to previous schemes.

KEYWORDS

Internet of Things; blockchain; forward-secure; blind signature

1 Introduction

The Internet of Medical Things (IoMT) has significantly transformed the traditional healthcare industry in recent years [1]. It comprises a range of smart medical devices that can sense medical data, along with transmitters that enable the secure transmission of sensitive information [2]. By interconnecting these devices and transmitters, IoMT facilitates real-time monitoring of the health status of patients. At the same time, the vast amount of sensitive medical data, such as electronic health records (EHRs), presents significant challenges for data security and privacy protection [3]. Therefore, IoMT incorporating blockchain technology has been proposed (e.g., [4–7]). The decentralized nature of blockchain and data immutability enhances the security of medical data.

However, traditional blockchain-enabled IoMT schemes [8–12] rely on public transaction records and digital signatures to ensure data integrity. Although medical information such as electronic medical records are uploaded to the blockchain after signing, they are still at risk of leakage as the data is transparent to the signer.

To overcome these obstacles, many scholars have applied blind signature technology to BIoMT [13–16], which allows users to sign without knowing the content of EHRs. Blind signatures protect the security of medical data and verify its integrity.

Unfortunately, these solutions are either ineffective against the risks associated with quantum computing attacks or face the potential for key compromise. Specifically, the key required for the user to sign in several schemes is constant after generation [13–17]. It means that if the key is compromised due to an external attack or improper storage, adversaries could potentially compromise the content of data previously signed by the signer. For instance, in the scheme [13], a trusted organization generates a pair of large numbers as public and private keys for each user. The user applies a blind signature to the data using their individual private key. However, the private key is retained by the user and remains unchanged throughout the duration of the scheme, which exposes it to the risk of key leakage [18]. On the other hand, as quantum computers advance, blind signature schemes reliant on the discrete-logarithm problem or number-theoretic hard problems, including Elliptic Curve Cryptography (ECC) [19] and Rivest-Shamir-Adleman (RSA) [12], will cease to be secure. The computational prowess offered by quantum computers can convert these hard problems into polynomial time-solvable ones through the application of Shor's algorithm [20].

We introduce lattice based forward-secure blind signature (LFSBS), an efficient lattice-based blind signature scheme with forward security for blockchain-enabled IoMT, aimed at enhancing the protection of sensitive medical data. The scheme is not only resistant to quantum attacks, but also supports forward security. LFSBS addresses two key challenges in blockchain-enabled IoMT, as outlined below.

The first challenge is to achieve quantum resistance for blind signatures. The security of lattice-based cryptosystems relies on the intractability of problems on the lattice, such as the least integer solution (SIS). These problems also have no significant computational advantage on quantum computers. Therefore, lattice-based cryptographic regimes are considered to be resistant to attacks on quantum computation. LFSBS constructs an anti-quantum blind signature protocol based on the lattice theory. In addition, computational complexity and communication efficiency are taken into account in the protocol design to ensure that the anti-quantum nature is satisfied while still maintaining efficient execution performance.

The second challenge is how to design a forward-secure key evolution mechanism, which provides strong forward security to ensure that past keys are not compromised by leakage of existing keys.

LFSBS assigns time periods to the leaf nodes of a binary tree. When the time period changes, the corresponding leaf nodes change accordingly. The *ExtBasis* algorithm [21] is then applied to update the keys associated with these leaf nodes.

In summary, the contributions of this work are shown below:

- We introduce a quantum-resistant forward-secure blind signature scheme, LFSBS, which realizes an attack on quantum computing based on the SIS assumption. In addition, we further design a forward secure key evolution mechanism for LFSBS to prevent key leakage.
- We formally define and prove the blindness and forward unforgeability of LFSBS in the Random Oracle Model (ROM). We show through comprehensive experimental results that LFSBS has significant efficiency advantages over previous techniques in signature generation and verification. For instance, when $\boldsymbol{n} = 25$, ***Sign*** and ***Verify*** in LFSBS are about 1.2×–3.4× and 3.5×–16× faster than other schemes, respectively. The total time reduction is 22%–73%.

2 Related Work

As blockchain technology rapidly advances and is widely adopted in the IoMT, privacy protection has increasingly become a significant concern. Blockchain offers a new approach for securely transmitting and storing medical data through its decentralized and tamper-proof characteristics [22]. However, IoMT devices face serious privacy challenges when collecting and transmitting sensitive health data. How to effectively protect user privacy while securing data has become an urgent challenge. Garg et al. [6] introduced a novel blockchain-based scheme for authentication and key management called BAKMP-IoMT, which ensures the reliability of medical data during transmission by establishing a secret pairing key between an individual, a medical device, and a server, but it involves a large number of key management issues. In [11], Rachakonda et al. introduced the SaYoPillow system, which designs secure data transmission, storage, and communication protocols for uploading and retrieval in order to reduce malicious attacks on medical data during interaction with the blockchain, but it suffers from deficiencies in data validity validation, which may affect its security and reliability in practical applications. Nie et al. [23] proposed a novel blockchain-assisted data transfer scheme in which Bloom filters with hash functions were designed to ensure data authenticity. Meanwhile, Bhattacharjya et al. [24] used elliptic curve digital signature for signing medical information to verify the reliability of data. However, in these designs, the signer has high visibility to the specific content of the data, which may affect the effectiveness of privacy protection. To further protect the data privacy of distributed ledgers in blockchain-enabled IoMT and improve the reliability of transactions, a privacy-preserving scheme utilizing ElGamal blind signatures was proposed by Le et al. [12]. Meanwhile, Li et al. [14] introduced the concept of 'swarm' on the basis of blind signatures, and implemented the mechanism of co-signing by multiple entities in the blockchain, which improves the security of the signing process.

However, none of the above schemes considered security under quantum attacks, Li et al. [25] then developed a blind signature scheme using lattice assumptions to counter quantum attacks. Qu et al. [26] designed a novel quantum blockchain-based system for medical data processing (QB-IMD). The system features a quantum blockchain framework that leverages quantum signatures and authentication to guarantee data integrity and protection against tampering. The above blind signature techniques, nevertheless, are still unable to cope with the risk of key leakage, leading to the fact that even with anti-quantum blind signatures to enhance data protection, the security of historical data may still be threatened in case of key leakage. Therefore, introducing forward security in blind signatures can effectively solve this problem by periodically updating the key or using a new key in each session to

ensure that past data remains secure even if the key is cracked in the future. Currently, most solutions [27–30] are based on number-theoretic assumptions or only support ordinary signatures, and thus lack security against quantum attacks.

3 Preliminaries

We present the preparatory knowledge involved in the program in this section.

3.1 Relevant Theory

Definition 1. Let the matrix $B = (b_1, \ldots, b_n) \in \mathbb{Z}^{m\times n}$, where the vectors are linearly independent of each other. The lattice with matrix B as the basis is $\Lambda(B) := \{\sum_{i=1}^{i=n} b_i x_i | x_i \in \mathbb{Z}\} \subseteq \mathbb{Z}^m$. (When vectors are linearly independent, it is the discrete points generated by them that cover the entire discrete space and uniquely represent any vector in the space).

Definition 2. Given a matrix $A \in \mathbb{Z}_q^{m\times n}$, a vector $u \in \mathbb{Z}_q^n$, the q-order integer lattice is defined as follows:

$$\Lambda_q^{\perp}(A) = \{x \in \mathbb{Z}^m | Ax \equiv 0 \, mod \, q\} \tag{1}$$

$$\Lambda_q^{u}(A) = \{x \in \mathbb{Z}^m | Ax \equiv u \, mod \, q\} \tag{2}$$

Definition 3 [31]. For any parameter $\sigma > 0$, $x \in \mathbb{R}^m$, there is a Gaussian distribution function centred on c as $\rho_{\sigma,c}(x) = \exp(-\pi \parallel x - c \parallel^2 / \sigma^2)$. Let $\Lambda \subseteq \mathbb{Z}^m$ be a lattice, the discrete Gaussian distribution on Λ is is given by $D_{\Lambda,\sigma,c}(x) = \dfrac{\rho_{s,c}(x)}{\rho_{\sigma,c}(\Lambda)}$.

Lemma 1 (Rejection Sampling) [16, *Lemma* 4.5]. Given a subset $V = \{\mathrm{v} \in \mathbb{Z}^m : \parallel \mathrm{v} \parallel \leq T\}$ and a real number $s = \omega(T\log\sqrt{m})$, Define a probability distribution $h: V \to \mathbb{R}$ on V. There is a universal constant $M = O(1)$ such that the statistical distance $\Delta(\mathcal{A}, \mathcal{B}) := 2^{-\omega(\log m)}/M$ between the outputs of the two algorithms is negligible.

($\mathcal{A}$) With probability $1/M$, output (z, v) where $\mathrm{v} \leftarrow h, \mathrm{z} \leftarrow \mathcal{D}_{\sigma}^m$.

($\mathcal{B}$) With probability $\min\left(\dfrac{\mathcal{D}_{\sigma}^m(\mathbf{z},)}{M\mathcal{D}_{\mathbf{v},\sigma}^m(\mathbf{z})}, 1\right)$, output (z, v) where $\mathrm{v} \leftarrow h, \mathrm{z} \leftarrow \mathcal{D}_{\mathbf{v},\sigma}^m$.

Moreover, the probability that $\mathcal{A}$ outputs something is at least $(1 - 2^{-\omega(\log m)})/M$. For any $\alpha > 0$ and $s = \alpha T$, the constant $M = e^{12/\alpha + 1/(2\alpha^2)}$, $\Delta(\mathcal{A}, \mathcal{B})$ is $2^{-100}/M$.

Lemma 2 [16]. For any $\mathrm{v} \in \mathbb{Z}^m$, if $s = \alpha \cdot \parallel \mathrm{v} \parallel$, where $\alpha > 0$, there exists a probability $\Pr\left[\mathcal{D}_s^m(\mathrm{x})/\mathcal{D}_{s,\mathrm{v}}^m(\mathrm{x}) \leq e^{12/\alpha+1/(2\alpha^2)} : \mathrm{x} \leftarrow \mathcal{D}_s^m\right] \geq 1 - 2^{-100}$.

3.2 Relevant Algorithm

TrapGen(n, m, q) [32]: Let n, $q = poly(n)$, and $m \geq 5nlogq$ be integers. There exists a polynomial-time algorithm *TrapGen* that outputs a pair of matrices $(A \in \mathbb{Z}_q^{n\times m}, T \in \mathbb{Z}^{m\times m})$, where A is uniformly distributed over $\mathbb{Z}_q^{n\times m}$, and T is a basis for $\Lambda_q^{\perp}(\mathrm{A})$ with the property that $\parallel T \parallel \leq O(\sqrt{n\log q})$.

ExtBasis $\left(A, T_{A_2}\right)$ [21]: Let a matrix $A = [A_1, A_2, A_3]$. If there exists T_{A_2} which is a basis for $\Lambda_q^{\perp}(A_2)$, then there exists a deterministic polynomial-time algorithm *ExtBasis* that outputs a basis T_A for $\Lambda_q^{\perp}(\mathrm{A})$ such that $\parallel \widetilde{T_A} \parallel = \parallel \widetilde{T_{A_2}} \parallel$.

SampleKey(A, T, r, K) [33]: Given the output (A, T) of the algorithm *TrapGen*, a real number $r > \| \tilde{\mathrm{T}} \|$, and a matrix $K \in \mathbb{Z}_q^{n\times k}$, there exists a polynomial-time algorithm *SampleKey* that outputs a random matrix $S \in \mathbb{Z}^{m\times k}$ such that each column $\mathrm{S}[j] \in D$: $\{\mathrm{s} \in \mathbb{Z}^m : \| \mathrm{s} \| \leq r\sqrt{m}]$ for all $j \in [k]$, and $A \cdot S = K(\mathrm{modq})$ with overwhelming probability.

commit (M, b_3) [12]: Given a message $M \in \{0,1\}^*$ and a stream of characters $b_3 \in \{0,1\}^n$, there exists an algorithm *commit* with statistical hiding and computational constraints that outputs a commitment string.

3.3 Difficult Assumptions in Lattice

Definition 4 ($\updownarrow_2 - \mathrm{SIS}_{q,n,m,\beta}$ problem) [16]. For a random matrix $\mathrm{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, find a vector $\mathrm{z} \in \mathbb{Z}^m\{0\}$ with $Az = 0\,(modq)$ and $\| \mathrm{z} \| \leq \beta$.

Lemma 3 [34]. For any $\mathrm{A} \in \mathbb{Z}_q^{n\times m}$, where $m > 64 + n \cdot \log q / \log(2d + 1)$, and any random vector $\mathrm{s} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^m$, the probability that there exists another vector $\mathrm{s}' \in \{-d, \ldots, 0, \ldots, d\}^m$ such that $\mathrm{As} = \mathrm{As}'$ is $1 - 2^{-100}$.

4 Framework Description

This section provides a formal description of the target problem addressed in this paper. We first briefly describe the system model for LFSBS. Following that, we present the LFSBS definition and security model. Details of the notations used can be found in Table 1.

Table 1: Notations

Symbol	Definition
dp	Binary tree depth
ρ	Number of time periods corresponding to tree depth
n	Security parameter
q, m, β	Lattice parameters
σ	Gaussian distribution parameter
$\sigma_1, \sigma_2, \sigma_3, M_i$	Rejected sampling parameters
k_1, k_2, γ	Parameters of the hash function with minimum entropy γ
pk, sk	Public and private keys

4.1 System Model

An LFSBS scheme consists of the following four entities, as depicted in Fig. 1. They are the smart medical devices in the perception layer, the accounting nodes and the blockchain-enabled IoMT in the network layer and the medical personnel in the application layer.

- **Smart Medical Devices:** Smart medical devices monitor and record the patient's physiological parameters, such as blood glucose and blood pressure. in real time through inbuilt sensors within the perceptual layer of the system. The raw medical data collected by these devices is initially processed to generate an electronic medical record and transmitted to the accounting node via a secure communication protocol.

- **Accounting Nodes:** The accounting node is responsible for ensuring the authenticity and privacy protection of the data. Upon receiving the electronic medical records, the accounting node will sign the data blindly to maintain privacy. It also ensures the non-repudiation of the signature due to the non-forgeability of the blind signature, which enhances the trustworthiness of the entire system. The master accounting node that obtains the right to establish the block broadcasts the signature data to other accounting nodes in the chain, which use the public key to verify the data with the signature. When all nodes have completed the verification, they return the results to the master accounting node, which eventually embeds the data in the transaction record and incorporates it into the blockchain network.
- **Medical Blockchain Network:** The medical blockchain network is responsible for storing and verifying the data. All signed data verified by the accounting node is recorded in the distributed ledger. The decentralised nature of the blockchain ensures that each piece of data cannot be tampered with after it is written and the integrity of the data is confirmed through a consensus mechanism.
- **Medical Personnel:** The data on the blockchain can be accessed by medical personnel through a permission management system for necessary analyses and decision making.

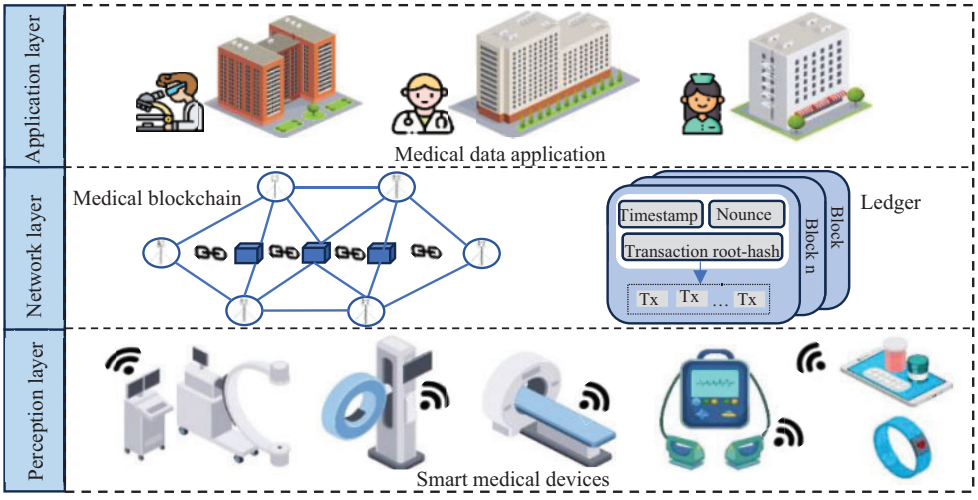


Figure 1: system model

4.2 The Syntax

A set of probabilistic polynomial time (PPT) algorithms $\prod$ LFSBS = (***Setup***, ***KeyGen***, ***Sign***, ***Verify***) are the foundation of an LFSBS scheme, and they are defined as follows:

- $(pp, pk, sk_0) \leftarrow Setup(1^{\lambda}, 1^{dp})$: is an algorithm executed by the authorised agency. It takes a security parameter λ and a binary tree depth dp as input. It outputs a public parameter pp, a public key pk, and an initial private key sk_0.
- $sk_{t+1} \leftarrow KeyGen(pp, sk_t, t)$: is an algorithm executed by the Key Generation Centre (KGC). It takes the public parameter pp, the binary tree depth dp, and a time period t as input. It outputs a private key sk_{t+1} for time period $t + 1$.
- $(v, S) \leftarrow Sign\,(pp, pk, sk_t, t, M)$: is an algorithm executed by the signer. It takes the public parameter pp, the public key pk, the private key sk_t, the time period t, and a message M as input. It outputs a signature pair (v, S).

- *value* $\leftarrow$ *verify*(pp, pk, t, S, M): is an algorithm executed by the verifier. It takes the public parameter pp, the public key pk, the time period t, the signature S, and the message M as input. It outputs a return value *value*.

4.3 Security Model

We introduce two security concepts for the LFSBS scheme: one is called blindness in order to ensure privacy during the signature process; the other is called forward unforgeability for the reliability of the signature.

4.3.1 Blindness

The unforgeability of LFSBS is determined by the interaction between $\mathcal{C}$ and $\mathcal{A}$, where $\mathcal{C}$ denotes the challenger and $\mathcal{A}$ denotes the attacker.

Setup (1^{λ}): The challenger $\mathcal{C}$ provides the attacker $\mathcal{A}$ with a public parameter pp, a public key pk, and a private key sk_0 using the initialization algorithm.

Challenge: The attacker $\mathcal{A}$ chooses two distinct messages M_1 and M_1 and submits them to the challenger $\mathcal{C}$. Subsequently, $\mathcal{C}$ randomly selects a bit b and engages with $\mathcal{A}$ as a signer, resulting in the generation of two pairs of signatures (v_b, S_b) and (v_{1-b}, S_{1-b}).

Output: $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

$\mathcal{A}$ will win the game described above if $b = b'$.

We define the advantage of the attacker in the blindness game as $Adv_{\mathcal{A}}^{Blindness} = \Pr[\mathcal{A}wins] - 1/2$.

Definition 1: If the $Adv_{\mathcal{A}}^{Blindness}$ can be ignored by any PPT attacker $\mathcal{A}$, then the LFSBS scheme is considered perfectly blind.

4.3.2 Forward-Secure Unforgeability

The unforgeability of LFSBS is determined by the interaction between $\mathcal{C}$ and $\mathcal{A}$, where $\mathcal{C}$ denotes the challenger and $\mathcal{A}$ denotes the attacker.

Setup (1^{λ}): The challenger $\mathcal{C}$ provides the attacker $\mathcal{A}$ with a public parameter pp and a public key pk using the initialisation algorithm.

Query: At a time period t, the attacker $\mathcal{A}$ can adaptively perform the following polynomial-time random queries.

1) Key oracle $\mathcal{KO}(t)$ query: The attacker $\mathcal{A}$ interacts with the random oracle $\mathcal{KO}$ to obtain the corresponding private key sk_{t+1} if $t < \rho - 1$; Otherwise, it returns an empty string.
2) Hash oracle $\mathcal{HO}(t, M)$ query. The attacker $\mathcal{A}$ interacts with the random oracle $\mathcal{HO}$ to obtain the corresponding hash value.
3) Signing oracle $\mathcal{SO}(t, M)$. query: The attacker $\mathcal{A}$ interacts with the random oracle $\mathcal{SO}$ to obtain corresponding signature.
4) Intrusion oracle $\mathcal{IO}(\tilde{t})$ query: The attacker $\mathcal{A}$ interacts with the random oracle $\mathcal{IO}$ to obtain corresponding private key $sk_{\tilde{t}}$ and transfer the gaming process to the output phase.

Output: $\mathcal{A}$ outputs a signature (v^*, S^*) with a message M^* at time period t^*.

$\mathcal{A}$ will win the game described above if $t^* < \tilde{t}$ and *verify* $(M^*, pk, S^*) = 1$.

We define $Adv_{\mathcal{A}}^{unforgeability} = \Pr[\mathcal{A}wins]$.

Definition 2: If the $Adv_{\mathcal{A}}^{unforgeability}$ can be ignored for any PPT attacker $\mathcal{A}$, then the LFSBS scheme is considered forward-secure unforgeable.

5 Our Proposed Scheme

We describe in detail the individual algorithms involved in the LFSBS scheme in this section.

5.1 System Initialization

The system is initialized by the authorized agency through the execution of the algorithm ***Setup***. Initially, the agency chooses prime numbers $q = ploy(n)$, $m = O(nlogq)$, k_1, k_2, $\rho = 2^{dp-1}$, σ ,...,σ_3, and randomly selects matrices $A_1 \leftarrow \mathbb{Z}_q^n$, $(A_1^0, A_1^1, \ldots, A_{dp}^0, A_{dp}^1) \leftarrow \mathbb{Z}_q^{n\times m}$. To ensure system security within random oracles, the agency picks a hash function: H: $\{0,1\}^* \rightarrow \{e_2 \in \{-1,0,1\}^{(dp+1)m}, ||e_2|| \leq k_2\}$, where H is a one-way hash function that maps strings to a vector e_2. Additionally, the agency defines the commit function $commit$: $\{0,1\}^* \times \{0,1\}^n \rightarrow \{0,1\}^*$ and executes the algorithm $TrapGen(n,m,q)$, to acquire a pair of matrices (A, T_A), where $A \in \mathbb{Z}_q^{n\times m}$ and $T_A \in \mathbb{Z}_q^{m\times m}$ is the basis for $\Lambda_q^{\perp}(A)$ satisfying $||\widetilde{T_A}|| \leq O(\sqrt{n\log q})$. The public key pk is the set of matrices $(A, A_1^0, A_1^1, \ldots, A_{dp}^0, A_{dp}^1, A_1)$, and the private key sk_{root} is T_A. The system's public parameters pp is generated as $pp = (q, m, n, k_1, k_2, \rho, \sigma, \ldots, \sigma_3, H, commit)$.

5.2 Key Generation

The KGC obtains the keys of all internal nodes that can generate previous keys by executing the algorithm ***Keygen***, which utilizes the key evolution mechanism.

Each time period ρ corresponds in order from left to right to a leaf node l in a binary tree of depth dp. For any leaf node l, there is a minimal subset of nodes, denoted as $Note(l)$, that includes at least one ancestor node common to all leaves from l to the last leaf $\rho - 1$, while excluding any ancestor node of any leaf from 0 to $l - 1$. Such as using binary codes for node names as illustrated in Fig. 2. $Note\,(l = 0) = \{root\}$, $Note\,(l = 1) = \{01, 1\}$, $Note\,(l = 2) = \{1\}$, $Note\,(l = 3) = \{11\}$. The private key sk_t, corresponding to time period t, consists of the private keys of all nodes in the minimal subset $Note(l = t)$. As shown in Fig. 2, $sk_{root} = sk_0 = \{T_A\}$, $sk_1 = \{T_{01}, T_1\}$, $sk_2 = \{T_1\}$, $sk_3 = \{T_{11}\}$, where T_{01}, T_1, and T_{11} are trapdoors associated with $[A||A_1^0||A_2^1]$, $[A||A_1^1]$ and $[A||A_1^1||A_2^1]$, respectively.

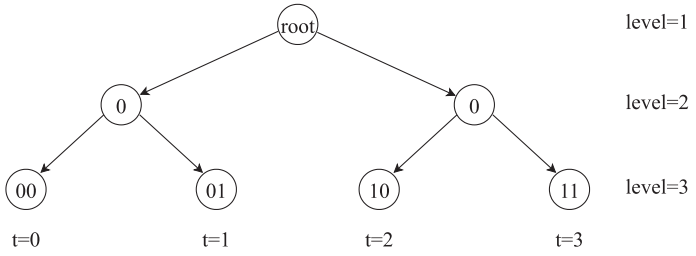


Figure 2: Binary tree with depth $dp = 2$ and time period $\rho = 4$

The update from the private key sk_t to sk_{t+1} using the trapdoor delegation mechanism with the algorithm $ExtBasis$, consider a leaf node $l(i)$ with binary representation $\{0,1\}^i$. The matrix corresponding to the private key $sk_{l(i)}$ is $T_{l(i)}$, which is computed from the initial private key T_A via $ExtBasis(A_{l(i)}, T_A)$, where $A_{l(i)} = [A||A_1^{l(i)}||\cdots||A_i^{l(i)}]$. Therefore, if the private key $T_{l(j)}$ of any ancestor

node from time period $t(i)$ is known, along with the associated matrix $A_{t(j)}$, it is possible to derive the private key of $T_{t(i)}$, provided that $t(j) < t(i)$.

5.3 Signature Generation

The accounting node acts as a signer generates the signature by interacting with the user executing the algorithm ***Sign***. The interactive process is as follows:

Step 1: The signer first constructs a matrix $A_{t(i)} = [A||A_1^{t(i)}|| \cdots ||A_{dp}^{t(i)}] \in \mathbb{Z}_q^{n\times(dp+1)m}$ for time period $t(i)$. The signer then uses the algorithm *SampleKey* to obtain a temporary private key SK_t, which satisfies $A_{t(i)} \cdot SK_t = A_1$. Next, the signer calculates $x \in \mathbb{Z}_q^n = A_{t(i)}r$ after sampling the vector $r \in \mathbb{Z}^{(i+1)\times m} \leftarrow D_{\sigma_2}^{(i+1)\times m}$, and sends x to the user.

Step 2: The user obtains the vector x and samples random vectors as follows: $b_1 \leftarrow D_{\sigma_3}^{(dp+1)m}$,$b_2 \leftarrow D_{\sigma_1}^{(dp+1)m}$, and $b_3 \leftarrow \{0,1\}^n$. The user then computes the hash value $h = H(d,c)$, where vector $d = A_{t(i)}b_1 + x(modq)$ and $c = commit(M, b_3)$. Rejection sampling is subsequently employed to obtain the blinded challenge $e = h + b_2$, which is then returned to the signer with probability $min\left\{1, \frac{D_{\sigma_1}^m(\mathrm{e})}{N_1 \cdot D_{\sigma_1,h}^m(\mathrm{e})}\right\}$.

Step 3: The signer computes the blind signature $s = SK_t + e + r$, employing rejection sampling to confirm that the distribution of s is consistent with that of r. The signature s is then sent back to the user with probability $min\left\{1, \frac{D_{\sigma_2}^{(dp+1)m}(\mathrm{s})}{N_2 \cdot D_{\sigma_2,SK_t\mathbf{e}}^{(dp+1)m}(\mathrm{s})}\right\}$.

Step 4: The user receives a blind signature s and employs rejection sampling to compute the corresponding unblinded signature $us = s + b_1$ with probability $min\left\{1, \frac{D_{\sigma_3}^{(dp+1)m}(\mathrm{us})}{N_3 \cdot D_{\sigma_3,\mathrm{s}}^{(dp+1)m}(\mathrm{us})}\right\}$, ensuring us and s remain independent of each other. If $\| us \| < \sigma_3\sqrt{(dp+1)m})$ holds true, the user returns the final signature $S = (b_2, b_3, h, us)$ to the signer, with $result = ''\text{valid}''$. Otherwise, it outputs $result = (b_1, b_2, h, c)$.

Step 5: If the $result = ''\text{valid}''$, output $(v = (r, e, s), S)$. Otherwise, the signer performs the following checks and re-executes the signature algorithm. First, calculates $d = A_{t(i)}b_1 + A_1b_2 + x(mod\ q)$ and $d' = A_{t(i)}b_1 - A_1h + A_{t(i)}s(mod\ q)$. Then, verifies whether $e - b_2 = h = H(d', c)$ and $\| \mathrm{s} + b_1 \| \geq \sigma_3\sqrt{(dp+1)m})$. If these conditions are satisfied, revert to Step 1.

5.4 Signature Verification

The validity of the signature S is checked by the verifier by executing the algorithm ***Verify***. Initially, the verifier constructs the matrix $A_{t(i)} = [A||A_1^{t(i)}|| \cdots ||A_{dp}^{t(i)}]$ associated with $t\,(i)$ and computes $h' = H\left(A_{t(i)}\,(us - b_2 - h) - A_1\,(mod\ q)\,, commit(M, b_3)\right)$. The verifier returns 1 if both $h = h'$ and $\| us \| \leq \sigma_3\sqrt{(1+dp)m}$ are satisfied; otherwise, it returns 0.

6 Security Analysis

In this section, we provide a comprehensive assessment of the security of the LFSBS scheme with respect to the following three aspects: correctness, blindness, and forward-secure unforgeability.

6.1 Correctness

According to Lemma 2, the probability of $D_s^m(\mathrm{x})/D_{s,\mathrm{v}}^m(\mathrm{x}) \leq e^{12/12+1/(2\cdot 12^2)}$ is guaranteed to be at least $1 - 1/2^{100}$. Lemma 1 shows that rejection sampling necessitates $D_s^m(\mathrm{x})/(N \cdot D_{s,\mathrm{c}}^m(\mathrm{x})) \leq 1$, which means

that this condition holds only if $N \geq e^{1+1/288}$. Therefore, the signature algorithm generates a valid signature at most e^3 times repeatedly. Given a valid signature $S = (b_3, h, us)$, we have:

$$\begin{aligned}
&\mathbf{A_{t(i)}}\ (\mathbf{us} - \mathbf{b_2} - \mathbf{h}) - \mathbf{A_1}\ (\mathbf{modq}) \\
&= \mathbf{A_{t(i)}s} + \mathbf{A_{t(i)}b_1} - \mathbf{A_{t(i)}b_2} - \mathbf{A_{t(i)}h} - \mathbf{A_1}\ (\mathbf{modq}) \\
&= \mathbf{A_{t(i)}}(\mathbf{SK_t} + \mathbf{e} + \mathbf{r}) + \mathbf{A_{t(i)}b_1} - \mathbf{A_{t(i)}b_2} - \mathbf{A_{t(i)}}(\mathbf{e} - \mathbf{b_2}) - \mathbf{A_1}(\mathbf{modq}) \\
&= \mathbf{A_1} + \mathbf{A_{t(i)}e} + \mathbf{A_{t(i)}r} + \mathbf{A_{t(i)}b_1} - \mathbf{A_{t(i)}b_2} - \mathbf{A_{t(i)}e} + \mathbf{A_{t(i)}b_2} - \mathbf{A_1}\ (\mathbf{modq}) \\
&= \mathbf{x} + \mathbf{A_{t(i)}b_1}\ (\mathbf{modq}) \\
&= \mathbf{d}
\end{aligned} \tag{3}$$

6.2 Blindness

Assuming that an adversary $\mathcal{A}$ is unable to discriminate between blind signatures created by different messages, the proposed technique fulfills blindness. H is a hash function resistant to collisions, while *commit* is a statistically hidden commitment function.

Proof. In the following, we employ users u_0 and u_1 as challengers $\mathcal{C}$ to interact with adversary $\mathcal{A}$ played by the signer.

Initialization. The adversary $\mathcal{A}$ selects the security parameter λ, the binary tree depth dp and executes the function *Setup* to obtain a public parameter pp along with the public and initial private keys.

Challenge: Adversary $\mathcal{A}$ initially selects time periods t_0 and t_1, where t_0 and t_1 are not necessarily distinct. The adversary then obtains private keys sk_{t_0} and sk_{t_1}, which correspond to the outputs of the algorithm ***KeyGen***, along with distinct messages M_0 and M_1. $\mathcal{A}$ relays these messages to challenger $\mathcal{C}$, who then randomly picks a bit $b \in \{0, 1\}$ and uses the algorithm ***Sign*** to interactively sign the messages with adversary $\mathcal{A}$, resulting in the signatures $(S_b = (b_{2b}, b_{3b}, h_b, us_b), v_b = (r_b, e_b, s_b))$ and $((S_{1-b} = \left(b_{21-b}, b_{31-b}, h_{1-b}, us_{1-b}\right), v_{1-b} = (r_{1-b}, e_{1-b}, s_{1-b}))$.

Analysis: The adversary $\mathcal{A}$ constructs the blind signatures (r_b, r_{1-b}) and (s_b, s_{1-b}) using the signature algorithm. Since these signatures are self-generated, they are disregarded in the analysis. Additionally, the rejection sampling ensures that (e_b, e_{1-b}) and (us_b, us_{1-b}) follow the same distributions $D_{\sigma_1}^{(dp+1)m}$, and $D_{\sigma_3}^{(dp+1)m}$ respectively. The random sampling of the blind factors, coupled with the hash function H possessing the one-way collision-resistant property, implies that the adversary $\mathcal{A}$ cannot extract valid information from (b_{2b}, b_{21-b}), (b_{3b}, b_{31-b}) and (h_b, h_{1-b}). Even if the adversary $\mathcal{A}$ receives the $result = (b_1, b_2, h, c)$ and restarts the signature process at Step 5, this does not increase the adversary's chances of winning the game. This can be attributed to the fact that the blind factors b_1 and b_2 are generated through fresh random sampling by the user, while c is obtained by statistically concealing the commitment function *commit*. In summary, adversary $\mathcal{A}$ is unable to establish a correlation between the signature and the message.

6.3 Forward-Secure Unforgeability

Assuming an adversary $\mathcal{A}$ can break the forward-secure unforgeability of the proposed scheme with non-negligible probability using randomized oracle queries, then a polynomial-time algorithm B will exist that can solve the $l_2 - \mathrm{SIS}_{q,n,(1+2dp)m}$ problem with probability: $\beta = max\{(\sigma_2 + 2\sigma_3)\sqrt{(1+dp)\,m}, 2\,(\sigma_3 + \sigma_1 + 1)\sqrt{(1+dp)\,m}\}$.

Proof. Consider an instance of the $l_2\_SIS_{q,n,(1+2dp)m}$ problem is given by the equation $A \cdot \mathrm{z} = 0 \bmod q$ with $\| z \| \leq \beta$, where $A = \left[A_0 || U_1^0 || U_1^1 || \cdots || U_{dp}^0 || U_{dp}^1\right] \in \mathbb{Z}_q^{n \times (1+2dp)m}$ and $z \in \mathbb{Z}_q^{(1+2dp)m}$. Algorithm B will succeed in solving this instance if it can find a non-zero integer vector z.

Initialization. Algorithm B sets a public parameter pp according to the initialization algorithm ***Setup***, and defines the public key as follows. For each $i \leq dp$, $A_i^{t_i^*} = U_i^{t_i^*}$, B obtains $(A_i^{\mathrm{b}}, T_{A_i^{\mathrm{b}}})$ by executing algorithm ***TrapGen*** for each bit b where $b \neq t_i^*$. Then B sets $A_1 = A_{t(i)} \cdot SK_t$, where $SK_t^* \leftarrow D_\sigma^{(1+dp)m}$, $A_{t(i)} = [A||A_1^{t(i)}|| \cdots ||A_{dp}^{t(i)}] \in \mathbb{Z}_q^{n\times(dp+1)m}$. Subsequently, B sends the public parameter pp and the public key pk to the adversary $\mathcal{A}$ while keeping SK_t^* secret. Finally, B maintains an initially empty table T for storing random predictor queries (d, c) and their corresponding hash value h, and prepares a set of values $\{h_1, \ldots, h_{iH}\} \leftarrow S_H$ in response to the hash queries.

Queries:

Algorithm B acts as a signer and interacts with adversary $\mathcal{A}$ to form a signature. Adversary $\mathcal{A}$ is allowed to perform these specific oracle queries to B.

Key oracle $\mathcal{KO}(t(i))$. If $t(i)^* \geq t(i)$, the query is aborted, where $t(j) = (t_1, \ldots, t_{dp})$. Otherwise, B identifies k_1 as the minimum index with $k_1 \leq i$ and $t_{k_1} \neq t_{k_1}^*$. Subsequently, B obtains the matrix $T_{t_{k_1}} \leftarrow \mathrm{ExtBasis}(A_{t(k_1)}, T_{A_{k_1}^{t_{k_1}}})$ associated with the private key $sk_{t_{k_1}}$, where $A_{t(k_1)} = [A||A_1^{t_1}|| \cdots ||A_{k_1}^{t_{k_1}}]$.

Hash oracle $\mathcal{HO}(d, c)$. Upon receiving a hash query request from adversary $\mathcal{A}$, algorithm B first checks if the hash value corresponding to (d, c) is already present in the table T. If it is, B sends corresponding hash value to $\mathcal{A}$. If not, B selects the first unused hash value h_i from the set $\{h_1, \ldots, h_{iH}\}$, sends h_i to $\mathcal{A}$, and updates table T at the same time.

Signing oracle $\mathcal{SO}(t(i), M)$. If $t(i)^* \neq t(i)$, algorithm B computes $T_{A_{t(i)}} \leftarrow ExtBasis(A_{t(i)}, T_{A_{k_1}^{t_{k_1}}})$ and $SK_{t(i)} \leftarrow SampleKey(A_{t(i)}, T_{A_{t(i)}}, \sigma, A_1)$, where $A_{t(i)} = [A||A_1^{t_1}|| \cdots ||A_{dp}^{t_{dp}}]$. Otherwise, B sets $SK_{t(i)^*} = SK_{t(i)}{}^*$.

Intrusion queries $(TQ(t(i)))$. If $t(i) < t(i)^*$, B aborts the query. Otherwise, B sets the intrusion time $\tilde{t}(i) \leftarrow t(i)$, calculates the corresponding private key $sk_{\tilde{t}(i)}$, and provides it to $\mathcal{A}$.

Output: $\mathcal{A}$ Output of falsified data: $(t_1(i)^*, M_1^*, S_1^* = (b_{3_1}^*, b_{2_1}^*, h_1^*, us_1^*))$, where $h_1^* = H(A_{t(i)^*}(us_1^* - b_{2_1}^* - h_1^*) - A_1(modq), commit(M_1^*, b_{3_1}^*))$. B accepts the signature if $t_1(i)^* = t(i)^*$.

Analysis: Let i denote the target fork index with $i \leq iH$ and $h_i = h_1^*$. B employs a backtracking strategy to retain the set $\{h_1, \ldots, h_{i-1}\}$ and selects a fresh set $\{h_i', \ldots, h_{iH}'\}$. Together, they form the returned set $\{h_1, \ldots, h_{i-1}, h_i', \ldots, h_{iH}'\}$ for the hash query. Similarly, $\mathcal{A}$ outputs a new signature $(t_2(i)^*, M_2^*, S_2^* = (b_{3_2}^*, b_{2_2}^*, h_2^*, us_2^*))$ via the above oracle, where $h_2^* = h_i'$. If $t_2(i)^* \neq t(i)^*$ or $h_2^* = h_1^*$, then B aborts. If $h_2^* \neq h_1^*$, B returns the data pair $(A_{t(i)^*}(us_1^* - b_{2_1}^* - h_1^*) - A_1(modq), commit(M_1^*, b_{3_1}^*), A_{t(i)^*}(us_2^* - b_{2_2}^* - h_2^*) - A_1(modq), commit(M_2^*, b_{3_2}^*))$. Additionally, we have $A_{t(i)^*}\tilde{v} = 0(modq)$, where $\tilde{v} = (us_1^* - us_2^* + b_{2_2}^* - b_{2_1}^* + h_2^* - h_1^*)$, since the data pair originates from the consent hash query and the commitment function is binding. According to Lemma 3, there exists at least one key SK' such that $A_{t(i)^*}SK^* = A_{t(i)^*}SK'$ with $SK' \neq SK^*$. Therefore, at least one $\tilde{v} \neq 0$ exists with $A_{t(i)^*}\tilde{v} = 0(modq)$. Note that $\| b_{2\{1,2\}}^* \| \leq \sigma_1\sqrt{(1+dp)m}$, $\| us_{\{1,2\}}^* \| \leq \sigma_3\sqrt{(dp+1)m}$, $||h_{\{1,2\}}^*|| \leq \sqrt{(dp+1)m}$. Therefore, $\| \tilde{v} \| \leq 2(\sigma_3 + \sigma_1 + 1)\sqrt{(1+dp)\,m}$.

In the following we consider the case where adversary $\mathcal{A}$ forges a signature by restarting the signature interaction. Specifically, when A returns $result = (b_1, b_2, h, c)$ to B, B can only restart the interaction with $\mathcal{A}$ if $h = \left(A_{t(i)^*}(us - b_2 - h) - A_1(modq), c\right)$. Suppose adversary $\mathcal{A}$ successfully produces a valid signature $\hat{S} = (\widehat{b_2}, \widehat{b_3}, \hat{h}, \widehat{us})$ after passing $\widehat{result} = (\widehat{b_1}, \widehat{b_2}, \hat{h}, \hat{c})$ to B. This implies that the following equation must hold: $e - \widehat{b_2} = \hat{h} = H\left(A_{t(i)^*}b_1 + x(modq), c\right) = H\left(A_{t(i)^*}(\widehat{us} - \widehat{b_2} - \hat{h}\right) -$

$A_1(modq), commit(M^*, \widehat{b_3}))$, where $\| \widehat{us} \| \leq \sigma_3\sqrt{(1+dp)m}$. If $\hat{h} \neq h$, then B aborts. Otherwise, we have $A_{t(i)^*}\widehat{us}(modq) = A_{t(i)^*}b_1 + A_{t(i)^*}s(modq)$, leading to $A_{t(i)^*}\hat{v}\,(modq) = 0$, where $\hat{v} = b_1 + s - \widehat{us} \neq 0$. If $\hat{v} = 0$, then $\| b_1 + s \| = \| \widehat{us} \| \leq \lambda\sigma_3\sqrt{m}$, which contradicts $\| s + b_1 \| \geq \sigma_3\sqrt{(dp+1)m)}$ from Step 5. Therefore, $\| \hat{v} \| \leq \| b_1 \| + \| s \| + \| \widehat{us} \| \leq (2\sigma_3 + \sigma_2)\sqrt{(1+dp)m}$. Finally, we obtain the matrix A by inserting the correlation matrix $U_i^{1-t_i^*}$ into the matrix $A_{t(i)^*}$. Similarly, the vector v is derived by inserting the element 0 into the corresponding position of the vector $\hat{v}$. Thus we obtain an instantiation of the SIS problem $l_2\_SIS_{q,n,(1+2i)m}$. The probability of solving it successfully is given by $\beta = max\{(\sigma_2 + 2\sigma_3)\sqrt{(1+dp)\,m}, 2\,(\sigma_3 + \sigma_1 + 1)\sqrt{(1+dp)\,m}\}$.

7 Performance Evaluation and Comparison

We conducted a thorough experimental evaluation of the LFSBS and compared its performance with other schemes. All experiments were carried out on a Windows 11 operating system using an 11th Gen Intel(R) Core(TM) i5-8250U @ 1.60 GHz processor and 12 GB of RAM. The LFSBS scheme was fully implemented in the Python. Finally, the relevant parameter settings involved in the experiment are presented in Table 2.

Table 2: Parameter configuration

Parameter	Definition	Sample
q	–	2^{27}
n	–	2^9
m	–	13,824
σ	$\geq O\left(\sqrt{n\log q}\right) \cdot \omega\left(\sqrt{\log n}\right)$	2^{11}
σ_1	$12\sqrt{k_2}$	2^6
σ_2	$12\sigma\eta\sigma_1\sqrt{(1+dp)\,mk_1}$	2^{20}
σ_3	$12\eta\sigma_2\sqrt{m}$	2^{30}
η	[1.1, 1.3]	1.1
k_2	$2^{k_2} \cdot \binom{k_1}{k_2} \geq 2^{100}$	28

7.1 Performance Evaluation

We assessed the effectiveness of our LFSBS scheme in this section, particularly focusing on the signature generation and verification algorithms, as these are the most time-consuming operations in LFSBS.

7.1.1 Computation Cost

We assessed the computational overhead associated with the **Sign** and **Verify** algorithms within LFSBS. Additionally, we evaluate the computational efficiencies of LFSBS with those of other schemes during the signing and verification processes, including cutting-edge lattice-based blind signature schemes [9,35,36]. As shown in Table 3.

Table 3: Comparison of computational overhead

Scheme	Sign	Verify
Yu et al. [35]	$5T_{add} + (\lambda + 3)T_{mul} + 4T_H + T_{Pis}$	$T_{add} + 4T_{mul} + T_H$
Alkadri et al. [36]	$(n+1)T_{add} + (\lambda + 3)T_{mul} + T_H + T_{Pis}$	$T_{add} + 2T_{mul} + T_H + T_{Pis}$
Alkadri et al. [9]	$(3k_1 + 4)T_{add} + 7k_1T_{mul} + T_H + 2T_{Pis}$	$2T_{add} + 2T_{mul} + T_H + T_{Pis}$
Our scheme	$5T_{add} + 4T_{mul} + T_H + T_{Pis}$	$T_{add} + 2T_{mul} + T_H$

Note: T_H represents the hash operation, T_{mul} represents the multiplication operation, T_{add} represents the addition operation, and T_{Pis} represents the preimage sampling.

7.1.2 Storage and Communication Overhead

In our LFSBS scheme, both public and private keys are composed of matrices. We evaluated the storage and communication costs of the **Sign** and **Verify** algorithms by determining the dimensions of the public key, private key and signature, as detailed in Table 4.

Table 4: Comparison of storage and communication overhead

Scheme	Public key	Secret key	Signature
Yu et al. [35]	$mn \log q$	$mnk_1 \log q$	$2nlogq$
Alkadri et al. [36]	$(2\lambda + k_1) \log q$	$(\lambda + k_1) \log q$	$(\lambda + k_1) \log q + \lambda dp$
Alkadri et al. [9]	$\lambda + m \log q$	$m(n+1) \log q$	$2\lambda + 3n \log q$
Our scheme	$n(k_1 + m(1 + 2dp))$ _	$m^2 \log q$	$n + k_1 + (dp + 1)m \log q$

7.1.3 Functional Evaluation

We compare four schemes in terms of blindness, quantum resistance and unforgeability, respectively in the Table 5. It's evident that our scheme offers functional advantages and improved feasibility.

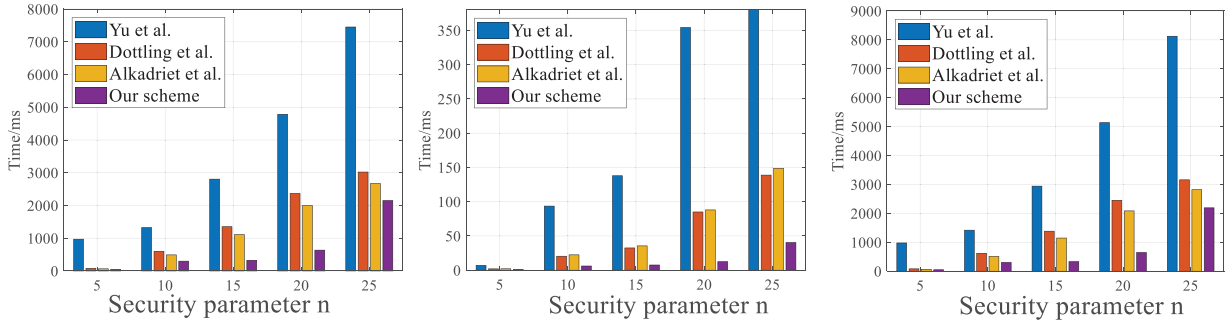
Table 5: Properties comparison with other schemes

Scheme	Blindness	Unforgeability	Quantum resistance	Forward security
Yu et al. [35]	✓	✓	✓	×
Alkadri et al. [36]	✓	×	✓	×
Alkadri et al. [9]	✓	✓	✓	×
Li et al. [14]	✓	×	✓	×
Crites et al. [37]	✓	✓	×	×
Xu et al. [38]	✓	✓	✓	×
Our scheme	✓	✓	✓	✓

7.2 Performance Comparison

We provide an analytical comparison of the performance with the current lattice-based blind signatures schemes [9,35,36] in this section.

The running times of the verification and signature algorithms for our LFSBS scheme and those suggested in [9,35,36] are shown in Fig. 3. As shown, our LFSBS scheme outperforms the others significantly. For example, when $n = 25$, the signature and verification times for our scheme are 2.15 and 0.04 s, respectively. In comparison, the signature and verification algorithms of the schemes proposed in [9,35,36] take the following times: [35] takes 7.45 s for signature and 0.67 s for verification, [36] takes 3.02 s for signature and 0.14 s for verification, and [9] takes 2.67 s for signature and 0.15 s for verification. Therefore, we conclude that our LFSBS scheme is approximately 1.2× to 3.4× faster for signature algorithms and 3.5× to 16× faster for verification algorithms compared to those in [9,35,36]. As shown by the total computation time in Fig. 3. our LFSBS scheme is more efficient in generating signatures and verifying them than other schemes. In addition, we introduce a forward-secure key evolution mechanism to further enhance the key security.



(a) Sign computation overhead (b) Verify computation overhead (c) Total computation overhead

Figure 3: Comparison of the computation overhead associated with the attributes between our LFSBS and current lattice-based blind signature schemes [9,35,36]

In summary, our LFSBS scheme exhibits higher efficiency in signature generation and verification compared to existing lattice-based blind signature schemes. In addition, the scheme is not only resistant to attacks from quantum computers, but also possesses forward security, which better ensures the long-term security of the system and the integrity of the data.

8 Conclusion

The LFSBS designed in this paper is an anti-quantum blind signature scheme with forward security for BIoMT, which provides blind signatures for auditing users in BIoMT to protect medical information. Meanwhile, we introduce forward security to prevent key leakage and rely on lattice cryptography to defend against potential attacks from quantum computing. Experimental results show that the overhead of the algorithms in this scheme is effective for medical data sharing scenarios. In the future, we would like to further experiment this scheme in real BIoMT environments to ensure its performance and stability in real applications.

Acknowledgement: The authors extend their gratitude to the members of the research group for their invaluable support.

Funding Statement: This work was funded by the Yunnan Key Laboratory of Blockchain Application Technology (202105AG070005, 202305AG340008) & YNB202301, NSFC (Grant Nos. 72293583, 72293580, 62476007, 62176273, 62271234), and the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications)

(SKLNST-2024-1-06), the Project of Science and Technology Major Project of Yunnan Province (202302AF080006), Open Foundation of State Key Laboratory of Public Big Data (Guizhou University) under Grant No. PBD2022-16, Double First-Class Project for Collaborative Innovation Achievements in Disciplines Construction in Heilongjiang Province under Grant No. GXCG2022-054.

Author Contributions: The authors confirm their contributions to the paper as follows: study conception, design, simulation, analysis, interpretation of results and draft manuscript preparation: Gang Xu, Xinyu Fan, Xin Liu; interpretation of results and draft manuscript preparation: Xiu-Bo Chen, Zongpeng Li, Yanhui Mao, Kejia Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data is unavailable due to the nature of this research; participants did not provide consent for their data to be publicly shared. Therefore, supporting data is not accessible.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] X. Chen, S. Xu, Y. He, Y. Cui, J. He and S. Gao, "LFS-AS: Lightweight forward secure aggregate signature for e-health scenarios," in *IEEE Int. Conf. Commun.*, Seoul, Republic of Korea, 2022, pp. 1239–1244.

[2] C. Li *et al.*, "Efficient privacy-preserving in IoMT with blockchain and lightweight secret sharing," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 22051–22064, 2023. doi: 10.1109/JIOT.2023.3296595.

[3] X. Chen, S. Xu, T. Qin, Y. Cui, S. Gao and W. Kong, "AQ-ABS: Anti-quantum attribute-based signature for EMRs sharing with blockchain," in *IEEE Wireless Commun. Netw. Conf. (WCNC)*, Austin, TX, USA, 2022, pp. 1176–1181.

[4] G. Xu *et al.*, "PPSEB: A postquantum public-key searchable encryption scheme on blockchain for E-healthcare scenarios," *Secur. Commun. Netw.*, vol. 2022, no. 1, 2022, Art. no. 3368819. doi: 10.1155/2022/3368819.

[5] J. Miao, Z. Wang, Z. Wu, X. Ning, and P. Tiwari, "A blockchain-enabled privacy-preserving authentication management protocol for Internet of Medical Things," *Expert. Syst. Appl.*, vol. 237, 2024, Art. no. 121329. doi: 10.1016/j.eswa.2023.121329.

[6] N. Garg, M. Wazid, A. K. Das, D. P. Singh, J. J. Rodrigues and Y. Park, "BAKMP-IoMT: Design of blockchain enabled authenticated key management protocol for internet of medical things deployment," *IEEE Access*, vol. 8, pp. 95956–95977, 2020. doi: 10.1109/ACCESS.2020.2995917.

[7] S. Xu, X. Chen, Y. Guo, S. Yiu, S. Gao and B. Xiao, "Efficient and secure post-quantum certificateless signcryption for internet of medical things," *Crypt. ePrint Arch.*, vol. 2024, 2024, Art. no. 965.

[8] S. F. Ahmed, M. S. B. Alam, S. Afrin, S. J. Rafa, N. Rafa and A. H. Gandomi, "Insights into Internet of Medical Things (IoMT): Data fusion, security issues and potential solutions," *Inf. Fusion*, vol. 102, 2024, Art. no. 102060. doi: 10.1016/j.inffus.2023.102060.

[9] N. A. Alkadri, Nabil, R. E. Bansarkhani, and J. Buchmann, "BLAZE: Practical lattice-based blind signatures for privacy-preserving applications," in *Int. Conf. Financial Cryptog. Data Secur.*, Kota Kinabalu, Malaysia, 2020, pp. 484–502.

[10] G. Xu, F. Yun, S. Xu, Y. Yu, X. Chen and M. Dong, "A blockchain-based log storage model with efficient query," *Soft Comput.*, vol. 27, no. 19, pp. 13779–13787, 2023. doi: 10.1007/s00500-023-08975-3.

[11] L. Rachakonda, A. K. Bapatla, S. P. Mohanty, and E. Kougianos, "SaYoPillow: A blockchain-enabled, privacy-assured framework for stress detection, prediction and control considering sleeping habits," 2020, *arXiv:2007.07377*.

[12] H. Le, D. H. Duong, and W. Susilo, "A blind ring signature based on the short integer solution problem," in *Inform. Secur. Appl: 20th Int. Conf., WISA 2019*, Jeju Island, Republic of Korea, Aug. 21–24, 2019, pp. 92–111.

[13] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, "PMRSS: Privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare," *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 1981–1990, 2021. doi: 10.1109/TII.2021.3070544.

[14] C. Li, B. Jiang, Y. Guo, and X. Xin, "Efficient group blind signature for medical data anonymous authentication in blockchain-enabled IoMT," *Comput. Mater. Contin.*, vol. 76, no. 1, pp. 591–606, 2023. doi: 10.32604/cmc.2023.038129.

[15] M. Bhavin, S. Tanwar, N. Sharma, S. Tyagi, and N. Kumar, "Blockchain and quantum blind signature-based hybrid scheme for Healthcare 5.0 applications," *J. Inf. Secur. Appl.*, vol. 56, 2021, Art. no. 102673. doi: 10.1016/j.jisa.2020.102673.

[16] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Annual Int. Conf. Theory Appl. Cryptograph. Tech.*, Cambridge, UK, Apr. 15–19, 2012, pp. 738–755.

[17] G. Xu *et al.*, "A model value transfer incentive mechanism for federated learning with smart contracts in AIoT," *IEEE Internet Things J.*, 2024. doi: 10.1109/JIOT.2024.3468443.

[18] S. Xu, Y. Cao, X. Chen, Y. Zhao, and S. Yiu, "Post-quantum public-key authenticated searchable encryption with forward security: General construction, and applications," in *Int. Conf. Inform. Secur. Crypt.*, Hangzhou, China, Dec. 9–10, 2023, pp. 274–298.

[19] S. H. Islam, R. Amin, G. P. Biswas, M. S. Obaidat, and M. K. Khan, "Provably secure pairing-free identity-based partially blind signature scheme and its application in online e-cash system," *Arab. J. Sci. Eng.*, vol. 41, no. 8, pp. 3163–3176, 2016. doi: 10.1007/s13369-016-2115-5.

[20] J. Howe, T. Pöppelmann, M. O'neill, E. O'sullivan, and T. Güneysu, "Practical lattice-based digital signature schemes," *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 3, pp. 1–24, 2015. doi: 10.1145/2724713.

[21] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Adv. Crypt.–EUROCRYPT 2010: 29th Annual Int. Conf. Theory Appl. Cryptog. Tech.*, Riviera, French, May 30–Jun. 3, 2010, pp. 553–572.

[22] X. Chen, S. Xu, Y. Cao, Y. He, and K. Xiao, "AQRS: Anti-quantum ring signature scheme for secure epidemic control with blockchain," *Comput. Netw.*, vol. 224, 2023, Art. no.109595. doi: 10.1016/j.comnet.2023.109595.

[23] X. Nie, A. Zhang, J. Chen, Y. Qu, and S. Yu, "Blockchain-empowered secure and privacy-preserving health data sharing in edge-based IoMT," *Secur. Commun. Netw.*, vol. 2022, no. 1, 2022, Art. no. 8293716. doi: 10.1155/2022/8293716.

[24] A. Bhattacharjya, K. Kozdrój, G. Bazydło, and R. Wisniewski, "Trusted and secure blockchain-based architecture for Internet-of-Medical-Things," *Electronics*, vol. 11, no. 16, 2022, Art. no. 2560. doi: 10.3390/electronics11162560.

[25] C. Li, Y. Tian, X. Chen, and J. Li, "An efficient anti-quantum lattice-based blind signature for blockchain-enabled systems," *Inf. Sci.*, vol. 546, no. 6, pp. 253–264, 2021. doi: 10.1016/j.ins.2020.08.032.

[26] Z. Qu, Y. Meng, B. Liu, G. Muhammad, and P. Tiwari, "QB-IMD: A secure medical data processing system with privacy protection based on quantum blockchain for IoMT," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 40–49, 2021. doi: 10.1109/JIOT.2023.3285388.

[27] M. Drijvers and G. Neven, "Forward-secure multi-signatures," *Crypt. ePrint Arch.*, vol. 2019, 2019, Art. no. 261.

[28] M. Abdalla, F. Benhamouda, and D. Pointcheval, "On the tightness of forward-secure signature reductions," *J. Cryptol.*, vol. 32, no. 1, pp. 84–150, 2019. doi: 10.1007/s00145-018-9283-2.

[29] Q. Xue, Z. Lu, and T. Zhang, "Attribute-based proxy signature scheme with dynamic strong forward security," *Int. J. Sens. Netw.*, vol. 44, no. 4, pp. 214–225, 2024. doi: 10.1504/IJSNET.2024.138518.

[30] J. Lee, J. E. Kim, and H. K. Oh, "Forward-secure multi-user aggregate signatures based on zk-SNARKs," *IEEE Access*, vol. 9, pp. 97705–97717, 2021. doi: 10.1109/ACCESS.2021.3093925.

[31] S. Xu *et al.*, "Lattice-based public key encryption with authorized keyword search: Construction, implementation, and applications," *Crypt. ePrint Arch.*, vol. 2023, 2023, Art. no. 1715.
[32] C. Ma, H. Gao, and B. Hu, "Ciphertext policy attribute-based encryption scheme supporting Boolean circuits over ideal lattices," *J. Inf. Secur. Appl.*, vol. 84, 2024, Art. no. 103822. doi: 10.1016/j.jisa.2024.103822.
[33] S. Xu, X. Chen, C. Wang, Y. He, K. Xiao and Y. Cao, "A lattice-based ring signature scheme to secure automated valet parking," in *Int. Conf. Wireless Algor., Syst. Appl.*, Nanjing, China, Jun. 25–27, 2021, pp. 70–83.
[34] H. Q. Le *et al.*, "Lattice blind signatures with forward security," in *Inform. Secur. Priv.: 25th Australasian Conf., ACISP 2020*, Perth, Australia, Nov. 30–Dec. 2, 2020, pp. 3–22.
[35] H. Yu and L. Baim, "Post-quantum blind signcryption scheme from lattice," *Front. Inform. Techn. Elect. Eng.*, vol. 22, no. 6, pp. 891–901, 2021. doi: 10.1631/FITEE.2000099.
[36] N. A. Alkadri, N. Dottling, and S. Pu, "Practical lattice-based distributed signatures for a small number of signers," in *Int. Conf. App. Crypt. Netw. Secur.*, Abu Dhabi, United Arab Emirates, Mar. 01, 2024, pp. 367–402.
[37] E. Crites, C. Komlo, M. Maller, S. Tessaro, and C. Zhu, "Snowblind: A threshold blind signature in pairing-free groups," in *Annual Int. Crypt. Conf.*, Santa Barbara, CA, USA, Aug. 09, 2023, pp. 710–742.
[38] G. Xu *et al.*, "A novel post-quantum blind signature for log system in blockchain," *Comput. Syst. Sci. Eng.*, vol. 41, no. 3, pp. 945–958, 2022. doi: 10.32604/csse.2022.022100.