# Secure Lattice-Based Signature Scheme for Internet of Things Applications

**SUNIL PRAJAPAT[1], (Student Member, IEEE), DEEPIKA GAUTAM[1], PANKAJ KUMAR[1], SRINIVAS JANGIRALA[2], ASHOK KUMAR DAS[3,4], (Senior Member, IEEE), AND BIPLAB SIKDAR[5], (Fellow, IEEE)**

[1]Srinivasa Ramanujan Department of Mathematics, Central University of Himachal Pradesh, Dharamshala 176206, India
[2]Jindal Global Business School, O. P. Jindal Global University, Sonipat, Haryana 131001, India
[3]Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad 500032, India
[4]Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul 02841, South Korea
[5]Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583

Corresponding authors: Ashok Kumar Das (iitkgp.akdas@gmail.com) and Pankaj Kumar (pkumar240183@gmail.com)

**ABSTRACT** The enormous benefits of Internet of Things (IoT) technology have driven its deployment in various applications. Additionally, the development of quantum computers has directed attention towards lattice-based cryptography. Consequently, the computational capabilities of quantum computers pose a threat to the security of the existing IoT signature mechanisms. Quantum computers are proficient at unraveling the complexity bound of computationally hard problems like the integer factorization problem (IFP) and the discrete logarithm problem (DLP). As a result, security is an essential requirement for the IoT communication network against quantum attacks. The amalgamation of certificateless public key cryptosystems (CL-PKC) and lattice-based cryptography (LBC) is one of the solution for alleviating these security menaces. Lucidly, CL-PKC prevents key escrow issues and key management problems; LBC prevents quantum attacks. The Shortest Integer Solution (SIS) problem, which the NTRU lattices offer, serves as the basis for this paper's introduction of a certificateless signature mechanism for IoT environments. By adopting the Random Oracle Model, we demonstrated the security of the suggested mechanism against Type 1 and Type 2 attackers. Furthermore, security analysis and performance evaluation demonstrate robust communication, as evidenced by metrics such as the computational cost of CL-Sign and CL-Verify phases at $536\mu s$, $376.81\mu s$ and communication cost of KGC at 418 bits, CL-Sign at 532 bits and CL-Verify at 446 bits. Also, we calculate the cost of single-message signature generation and verification on an IoT device. These results show that the suggested mechanism's security and computational efficiency are more reliable, and efficient than other relevant competing frameworks.

**INDEX TERMS** Certificateless public key cryptosystem, lattice-based cryptography, Internet of Things (IoT), signature, security.

## I. INTRODUCTION

As an unprecedented transformational shift, the Internet of Things (IoT) has a major function in the sustainable development of the computing economy. IoT has revolutionized technology and has driven it in a new direction. IoT interconnects smart objects with Internet connectivity and enables them with capabilities of producing, consuming, and exchanging data. These smart objects in IoT have limited computational capabilities and are entrenched with sensor chips and software [1]. Moreover, IoT is also referred to as *self-configuring universal network architecture*. The engrossment of IoT in daily routine has speeded up and simplified human life. Enormous advantages offered by the IoT have attracted the attention of researchers, paved the way for its deployment in various real-life applications

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamed Elhoseny.

such as healthcare, the environment, industries, and smart transportation, as enhanced in Fig. 1, and resulted in innovative services with amended flexibility and efficiency.

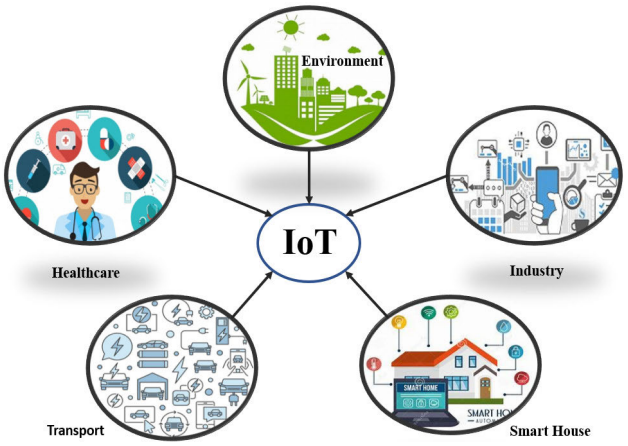


**FIGURE 1.** Varioius application areas of IoT.

With the accelerated progress of IoT networks, the number of connected devices and dependence on the daily routines is increasing rapidly. It has eventually enhanced the potential security perils in the network. The open communication network, limited resources, scalability, heterogeneity, and potential design deficiency are a few primary factors responsible for the IoT architecture's challenging security and privacy issues [2]. Provisions for, security and privacy in the communication network are obligatory. Security services such as non-repudiation, data integrity, confidentiality, and authentication are the necessities for secure communication and better deployment of the IoT. An adequate solution to the aforementioned issues is the combination of public key cryptography (PKC) and digital signatures. Numerous efficient public key cryptography-based digital signature mechanisms exist in the literature for the IoT. Regrettably, PKC faces a certification administration issues. The problem of the PKC is resolved with the ''Identity-Based Cryptography (IBC)'', presented by Shamir [3] in 1984. However, later it was observed that it also faces the key escrow problem.

Certificateless cryptography (CLC) is an optimum combination of PKC and IBC. CLC enables a user to produce private and public keys by himself. In CLC, a partial private key is invoked by a trusted third party, commonly referred to as the Key Generation Center (KGC), which the user later utilizes to calibrate his duo of keys. Consequently, the PKC and IBC problem are easily eradicated by the CLC. Unfortunately, the security of the existing digital signature mechanisms in the literature relies on the complexity assumptions of the Data Structures and Algorithms (DSA), bilinear pairing, and RSA, ElGamal, and Elliptic Curve Cryptography (ECC) based public key cryptosystems.

### A. MOTIVATION

Post-quantum computing has recently attracted attention with the quantum computer's fast establishment and development. Quantum computers are computationally proficient at unraveling the complexity bound of computationally hard problems like the ''Integer Factorization Problem (IFP)'' and ''Discrete Logarithm Problem (DLP)'' in polynomial time. Consequently, this has driven the security of the prevailing signature mechanisms of the IoT towards a critical reevaluation, and they are vulnerable to quantum attacks. Lattice-based cryptography (LBC) is an advanced approach with the potential to meet security requirements against quantum attacks. The fundamental computational assumptions include the shortest and closest vector problems, worst-to-average case reduction problems, ''Learning with Errors (LWE)'', and ''Ring Learning with Errors (RLWE)'' problems to ensure security in the lattice-based systems. An efficient signature mechanism with promising security is required for the deployment of the IoT. Additionally, mechanisms exploiting the advantages of LBC and certificateless cryptography can easily accomplish all the security requirements mentioned in an IoT architecture. Here is the motivation for this paper to design an efficient digital signature mechanism with the advantages of both LBC and CLC.

### B. RESEARCH CONTRIBUTIONS

The contributions of the proposed work are listed below:

1) This paper proposes a certificateless digital signature mechanism for an IoT network with the advantages of LBC (LB-CLS, in short). Therefore, the presented scheme can withstand quantum attacks.
2) The computationally hard SIS problem over NTRU assures the security of the proposed mechanism. Also, the security of the mechanism is demonstrated under the Random Oracle Model (ROM) against two adversaries: Type 1 and Type 2. Note that a Type 1 adversary is a hostile adversary who cannot access the system's primary key, but he/she can substitute a user's public key with any random value. Type 2 adversary cannot replace the user's public key, but he/she can access the private key of the Key Generation Center (KGC).
3) Performance analysis demonstrates robust communication, as evidenced by metrics such as the computational cost of CL-Sign and CL-Verify phases at $536\mu s$, $376.81\mu s$ and 376.81 mu/s and the communication cost of KGC at 418, CL-Sign at 532 and CL-Verify at 446 of the proposed mechanism against various existing mechanisms in the literature.

### C. PAPER ORGANIZATION

The paper's structure is as follows: Section II describes the related existing work. Section III presents the preliminaries, which contain the important definition and hardness of lattices. Section IV contains the structure and security model. Section V has the proposed work's construction, correctness, and security analysis. Section VII presents the computational

assessment of the proposed work. Finally, Section VIII concludes the paper.

## II. RELATED WORKS

Numerous authentication, encryption, and signature mechanisms have been presented in the literature. However, these mechanisms are vulnerable to quantum attacks. In 1996, Shamir [3] built the first lattice-based algorithm utilizing the lattice-based SIS hard problem. Alkim et al. [10] utilized the benefits o f L WE p roblems o ver l attices a nd projected a lattice-based signature mechanism called TESLA. Gupta and Biswas [11] put forward a lattice-based encryption and signature mechanism based on the ElGamal cryptosystem that uses the SIS problem for security.

Ghazinour et al. [12] put forward a lattice-based signature framework applicable to the embedded systems. El Moustaine and Laurent [13] proposed an authentication protocol for Radio Frequency Identification (RFID), w hose security relies on the NTRU lattices. Abdallah and Shen [14] put forward an abelian privacy preserving lattice-based system for the smart grid customer-side networks. They revealed that their proposed design contributes to the reduction of communication burden and preserves the user's privacy in the smart grid environment. Similarly, Güneysu et al. [12] came up with the lattice relying on a signature mechanism.

Ducas et al. [15] constructed a lattice relying on a signature mechanism by employing the modified rejection sampling technique. They also proposed a new rejection sampling algorithm with a bimodal Gaussian distribution. Implementation of the "Bimodal Lattice Signature Scheme (BLISS)" [15] on ARM Cortex-M4F microcontroller of 32-bit is presented by the Oder et al. [16] and they also investigated sampling techniques such as Ziggurat, Knuth-Yao, and Bernoulli. Xie et al. [4] presented a lattice-based signature scheme by utilizing the characteristics of the identity-based cryptosystem. The SIS problem assumption of the NTRU lattices provides the security of the mechanism. Similarly, Wu and Huang [5] developed an efficient identity-based forward signature mechanism. They employed the extended Samplepre and Lyubahevsky's methods in the proposed mechanism with the SIS problem of the lattice.

All the above-mentioned mechanisms rely either on PKC or IBC, and face the certificate supervision and key escrow issues. Therefore, these issues were resolved by Al-Riyami and Peterson [17] in 2003. Tian and Huang [6] proposed a certificateless s ignature s cheme e mploying L BC. Later, by employing the security offered by the SIS problem of NTRU lattices, Xie et al. [4] also put forward signature mechanism with the certificateless c ryptosystem. A reasonable high speed, low memory requirements and easily created short keys are some features offered by a new public key cryptosystem NTRU [18]. Hung et al. [7] came up with a revocable certificateless signature mechanism by employing the NTRU lattices. They constructed this mechanism to revoke illegal or malicious users through the revocation method. Regrettably, Shim [19] proved the vulnerabilities of Xie et al.'s scheme [4] and Hung et al.'s scheme [7] in opposition of the Type 1 and Type 2 adversaries. Xu et al. [8] presented a certificateless signature mechanism based on NTRU lattices for medical cyber-physical systems.On the other hand, a threshold signature with the certificateless cryptosysytem employing the SIS problem of the lattice against quantum attacks is presented by the Yu and Zhang [9]. The US National Institute of Standards and Technology (NIST) launched a "competition" in 2017 to develop a standard for digital signatures and quantum-safe key exchange. As of the time of writing, there are still two lattice-based signals in this process, both of which are based on the high-level designs mentioned above. It is currently in its third iteration. The Schnorr framework is followed by the CRYSTALS-Dilithium [20] system, however in order to maintain modest coefficient sizes, a critical rejection-sampling phase is included. Using a secret trapdoor for $f^{-1}$, the FALCON system [21] employs a randomized trapdoor sampling technique to generate *random* pre-images from a specific distribution. It is also essential to have the property that the distribution of the outputted pre-images does not leak trapdoor information, as there is no longer a bijection. The public key and signature size of both systems are the shortest of all quantum-safe signature schemes, and they are both comparatively quick.

Despite having a lattice foundation, the two systems differ greatly in their features. Although FALCON has fairly few parameters, the process of generating signatures with it is rather involved. It specifically makes use of the GPV sampler [22], which necessitates floating-point arithmetic with a precision of roughly 64 bits. Due to the high precision requirements, even with thorough testing, little implementation flaws could go undetected. Dilithium, on the other hand, is significantly less prone to implementation errors because it has greater parameters but a fairly straightforward implementation where all of the sampling in the signing is done in a power-of-2 range.

The techniques and highlights of the above-discussed schemes are summarized in Table 1.

## III. MATHEMATICAL PRELIMINARIES

Here, we go over the basics of mathematical preliminaries required to explain and evaluate the proposed scheme.

### A. LATTICE

The plenteous elements are denoted as follows. Let $V$ be a matrix and $\mathbf{v}$ be a vector. Then, $||\mathbf{v}||$ and $||\mathbf{v}||_\infty = \max[||\mathbf{v}_n||]$, respectively, denote the Euclidean norm of $\mathbf{v}$ and the prolonged norm of the whole columns of $\vec{\mathbf{v}}$. Assume that, $a = \sum_{n=0}^{N-1} a_n v_n$ and $b = \sum_{n=0}^{N-1} b_n v_n$ be two polynomials in $R_p$, where $R_p = \frac{Z_p}{(X^N+1)}$ denotes a polynomial ring with modulo $X^N + 1$ attended by quantum in $Z_p$, and $Z_p$ is the set of integers in the interval $(\frac{-p}{2}, \frac{p}{2}]$, for a prime $p$. According to

**TABLE 1. Techniques and highlights of various existing schemes.**

| Schemes | Techniques used | Highlights |
|---|---|---|
| Xie *et al.* [4] | Identity-based signature | security is demonstrated under random oracle model based on NTRU lattices assumptions and has key escrow problem. |
| Wu *et al.* [5] | Identity-based signature | Employed the extended Sample-pre and Lyubahevsky's methods with the SIS problem of the lattice and has key escrow problem. |
| Tian and Huang [6] | Certificateless and identity-based signatures | Proposed signature schemes are demonstrated secure under random oracle and designed by utilizing the NTRU lattice problem. |
| Hung *et al.* [7] | Certificateless signature | Proposed revocable signature scheme using NTRU lattices can not withstand Type 1 and Type 2 adversaries. |
| Xu *et al.* [8] | Certificateless signature | Proposed signature scheme for the medical cyber-physical system is based on the NTRU lattice. |
| You *et al.* [9] | Certificateless signature | Proposed a threshold signature scheme using inhomogeneous SIS problem and is shown to be secure against chosen-message attacks. |

the symbol $y \leftarrow S$, $y$ is uniformly chosen randomly from a set $S$. If there is a distribution $D$, the notation $z \leftarrow D$ indicates that $z$ was picked following the distribution $D$. A cross-section is a framework of focuses, and an $N$-sized lattice is a finished position discrete subgroup of $R_p$. Here, we consider the NTRU cross-sections.

*Definition 1 (Lattice):* Let $n$ vectors $\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_n$ be linearly independent and $B = \{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_n\}$ be a basis of the $n$-sized lattice $\Lambda$ generated along $B$. Then, $\Lambda$ can be denoted as $\Lambda = L(\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_n) = \{\sum_{n=1}^{N-1} x_n f_n : x_n \in R_p\}$.

*Definition 2 (NTRU Lattice):* Let $N$ be a power of 2 and $p$ be a prime. Additionally, $a, b \in R_p$, where $a$ is invertible modulo $p$. Assume that $h = b * a^{-1} \pmod{p}$. The NTRU that corresponds to the values of $h$ and $p$ is defined by $\Lambda_{h,p} = \{(u, v) \in R_p^2 \,|u + v * h = 0 \pmod{p}\}$. Here, the complete-rank lattice of $R^{2N}$ is created with a row of

$$F_{h,p} = \begin{bmatrix} -C_n(h) & I_n \\ pI_n & 0_n \end{bmatrix}$$

where $I_n$ and $0_n$ are square $N$-dimensional unitary and null-matrices, respectively, and $C(h)$ is a square $N$-dimensional anti-circulant matrix with $h$.

The basis $\Lambda_{h,p}$ is unsuitable for solving the standard lattice problem because $h$ is evenly distributed in $R_p$. As a result, Hoffstein et al. [23] solved the problem by converting another acceptable basis for $\Lambda_{h,p}$, namely,

$$B_{a,b} = \begin{bmatrix} C(b) & -C(a) \\ C(B) & -C(A) \end{bmatrix}$$

where $A, B \in R_p$ such that $a * B - b * A = p$. It can be calculated systematically to find $A$ and $B$. Moreover, $\vec{B_{a,b}}$ provides a short basis for $\Lambda_{h,p}$ because $||\vec{B_{a,b}}|| < ||F_{h,p}||$.

### B. ANTICIRCULANT MATRICES

Anticirculant matrices have unique compositions and functionality. We now define an $N$-dimensional anticirculant matrix $C_N(a)$ as follows [7].

*Definition 3: Let $C_N(a)$ be a Toeplitz-matrix. Then, it is represented by*

$$C_N(a) = \begin{bmatrix} (a) \\ (v.a) \\ . \\ . \\ . \\ (v^{N-1}.(a) \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & \ldots & a_{N-2} & a_{N-1} \\ -a_{N-1} & a_0 & \ldots & a_{N-3} & a_{N-2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ -a_1 & -a_{N-2} & \ldots & -a_{N-1} & a_0 \end{bmatrix}$$

*where $a = \sum_{n=0}^{N-1} a_n v_n \in R_p$. $C_N(a)$ is condensed as $C(a)$ in the sequence for convenience. The anticirculant matrices possess successive acceptable characteristics.*

### C. GAUSSIAN ON LATTICE

Gaussian sampling was initially offered to use a concise root as a trapdoor unescorted, expressive any information about it. On a lattice, the discrete Gaussian distribution is elucidated as follows.

*Definition 4 (Discrete Gaussian Distribution): For any $c \in R^N$ and any positive $s > 0$, the Gaussian function centered at c with a deviation variable s is elucidated as an $N$-sized Gaussian function $G_{s,c} : R^N \to (0, 1]$, $\forall x \in Z^N$ [24]:*

$$G_{s,c}(x) = \exp(-\pi \frac{||x - c||^2}{s^2}),$$
$$D_{s,c}(\Lambda) = \sum_{x \in \Lambda} G_{s,c(x)}.$$

*Here, the set of real numbers is R, and the set of integers is Z. For a natural number $N$, $R^N$ denotes an $N$-dimensional vector on real numbers field, and $Z_p^{m \times n}$ indicates a matrix of size $m \times n$ on the finite field $Z_p$. In an n-dimensional lattice $\Lambda$, we define the discrete Gaussian distribution as $\forall x \in \Lambda$:*

$$D_{\Lambda,s,c}(x) = \frac{G_{s,c}(x)}{G_{s,c}(\Lambda^u p(F))}.$$

*The distribution $D_{\Lambda,s,c}(x)$ is sufficiently defined over the lattice $\Lambda_p^{\perp}(F)$ for a matrix $F \in Z_p^{N*N}$ as $\Lambda_p^u(F)$. Then,*

$\forall x \in \Lambda$:

$$D_{\Lambda_p^u(F),s,c}(x) = \frac{G_{s,c}(x)}{G_{s,c}(\Lambda_p^u p(F))}.$$

*Lemma 1: Assume that $\Lambda$ is an $N$-sized lattice, and $p$ a prime. $B_{a,b}$ is a small basis if $s \geq ||\vec{B}_{a,b}||\omega(\sqrt{\log N})$ and $0 < \epsilon < 1$, where $\vec{B}_{a,b}$ indicates $B_{a,b}$'s Gram-Schmidt orthogonalization. Furthermore, we have the following [25]:*

$$D_{\Lambda,s,c}(x) \leq \frac{(1+\epsilon).2^{-n}}{1-\epsilon}$$

*has a part of $x \in R^N$ and $x \leftarrow D_{s,c}(\Lambda)$, and there is an algorithm, Gaussian Sampler, that creates a distribution statistically precise to $D_{s,c}^N$.*

*Lemma 2: For positive integer $N$ and any value of $s > 0$:*

*i) $Pr[x \leftarrow D_{z,s} : |x| > 12s] < 2^{-100}$,*

*ii) $Pr[x \leftarrow D_{z^N,s} : ||x|| > 2s\sqrt{N}] < 2^{-N}$.*

*Lemma 3: Let $\alpha$ be a positive real. Then, for any $v \in Z^N$, if $s = \omega(||v||\sqrt{\log N})$, then*

$$Pr[x \leftarrow D_{z^N,s} : \frac{D_{z^N,s}(x)}{D_{z^N,s,v}(x)} = 0(1)] = 1 - 2^{-\omega(\sqrt{\log N})},$$

*and more especially, if $s = \alpha||v||$, then*

$$Pr[x \leftarrow D_{z^N,s} : \frac{D_{z^N,s}(x)}{D_{z^N,s,v}(x)} < \exp^{12/a+1/(2a^2)}] = 1 - 2^{-100}.$$

We now describe the NTRU lattice's preimage sampling in Algorithm 1. The Gram-Schmidt orthogonalization of $B$ is represented in this sampling technique by the expression $SampleZ_{(s,c)}$ that samples a 1-dimensional Gaussian $G_{z,s_1,c_1}$ and $\vec{B} = (b_n)_{n\in N}$.

**Algorithm 1** GaussianSampler($B$, $s$, $c$) [4]

**Input:** $B$: basis, $s > 0$: standard deviation, $c \in z^N$: center of an $N$-dimensional lattice $\Lambda$.
**Output:** $k \in D_{\Lambda,s,c}$.

1: Set $k_n \leftarrow 0$
2: Set $c_n \leftarrow c$
3: **for** $n \leftarrow N, \ldots, 1$ **do**
4: Caculate $c_n \leftarrow < c_n, b_n > /||\vec{B}_n||^2$
5: Compute $s_n \leftarrow s/||\vec{B}_n||^2$
6: Compute $z_n \leftarrow SampleZ_{(s,c)}$
7: Calculate $c_{n-1} \leftarrow c_n - z_n b_n$ and $k_{n-1} \leftarrow k_n + z_n b_n$
8: **end for**
9: **return** $k_0$

### D. REJECTION SAMPLING TECHNIQUE

In lattice-based cryptography, Lyubashevsky [26] proposed a rejection sampling technique to sign a message. This approach is simple and needs just a few rejection samplings and matrix-vector multiplications. Making the distribution of output signatures independent of the signing key is the primary objective of a signature design rejection sampling technique. Furthermore, compared to Micciancio and Peikert's schemes [27], Lyubashevsky's scheme has a smaller signature and private key sizes while maintaining the same level of security. The primary distinction is that Lyubashevsky's scheme uses rejection sampling as an alternative to the hash-and-sign technique to create a signature. Algorithm 2 explains the rejection sampling approach.

**Algorithm 2** Rejection Sampling

**Input:** Cryptographic hash function $H : \{0,1\}^* \rightarrow \{v : v \in (-1,0,1)^k, ||v|| \leq c\}$ ($c$ is a constant, $k$ is a positive numer, and $k \ll p$, $m$ is the message, $W$ is a matrix which is arbitrarily chosen from $Z_p^{N*N}$, and $S$ is a signature key from $\{-d, \ldots, 0, \ldots, d, \ldots\}^{N*k}$.
**Output:** $z$ and $e$ vectors.

1: Sample $D_s^N$ arbitrarily to obtain $y$
2: Compute $c \leftarrow H(Wy, m)$
3: Compute $z \leftarrow sc + y$
4: Set $(z, e)$ with the probability min $\left\{1, \frac{D_s^N}{(WD_{Sc,s}^N(z)}\right\}$
5: **return** $(z, e)$

### E. HARDNESS HYPOTHESIS

This section addresses the SIS problem using lattices as the security postulates. The worst case of the "short independent vector problem (SIVP) with the approximation polynomial factor" is harder to solve than the SIS problem. The SIS problem and its premise are given below.

*Definition 5: Let $p$ and $\alpha$ be a certain whole number and a genuine number, respectively, and $\{a_1, a_2, \ldots, a_n\}$ be polynomials chosen consistently and separately from $R_p$. The $SIS_{p,n,\alpha}$ problem over grids is to find $n$ non-zero numbers $\{f_1, f_2, \ldots, f_n\}$ that fulfill two conditions [28]:*

*i).* $\sum_{n=1}^{N} a_n f_n = 0 \pmod{q}$,

*ii).* $||(f_1, f_2, \ldots, f_n)|| \leq \alpha$.

*Definition 6: Given a positive number alongside a genuine number $\alpha$ and $n$ polynomials $\{a_1, a_2, \ldots, a_n\}$ picked consistently and freely from $R_p$. Then, no probabilistic polynomial time (PPT) adversary $\mathcal{A}$ with an important likelihood of settling the SIS problem exists. The success probability (advantage) $ADV_A$ of the adversary $\mathcal{A}$ is then $ADV_A = Pr[A(\langle p, a_1, a_2, \ldots, a_n\rangle) = (f_1, f_2, \ldots, f_n) : ||(f_1, f_2, \ldots, f_n)|| \leq \alpha]$.*

The NTRU lattice's trapdoor generation technique differs slightly from conventional lattices and is referred to as trapdoor generation, which is provided in Algorithm 3.

## IV. STRUCTURE AND SECURITY MODEL FOR LATTICE-BASED CERTIFICATELESS SIGNATURE (LB-CLS)

This section provides the basic structure of the proposed generalized lattice-based certificateless signature scheme (LB-CLS). Next, we provide the security model associated with LB-CLS.

**Algorithm 3** TrapdoorGeneration $(N, p)$

**Input:** $N, p \in Z, s > 0$.
**Output:** $(B, h) \in R^{2N*2N} \times R_p^*$.

1: Sample $a$ and $b$ from $D_{Z^N,s}$ that satisfy $(a \bmod p) \in R_p^*$ and $(b \bmod p) \in R_p^*$.
2: **if** $(||a|| > s\sqrt{N}$ and $||b|| > s\sqrt{N})$ **then**
3: Repeat from Step 1.
4: **end if**
5: **if** $< a, b > \neq R$ **then**
6: Repeat from Step 1.
7: **end if**
8: To calculate $A_1, B_1 \in R$ such that $aB_1 - bA_1 = 1$, set $A_p = pA_1$ and $B_p = pB_1$.
9: Use the Babai's closest plane algorithm [29] to approximate the pair $(A_p, B_p)$ by a linear combination of $(a, b)$, $\ldots$, $(x_{N-1}a, x_{N-1}b)$. Let $(A, B)$ be an output such that $\exists$ $k \in R$ with $(A, B) = (A_q, B_q) - k(a, b)$.
10: **if** $||(A, B)|| > Ns$ **then**
11: Repeat from Step 1.
12: **end if**
13: Compute trapdoor basis $B = \begin{bmatrix} C(a) & C(b) \\ C(A) & C(B) \end{bmatrix}$ and polynomial $h = b * a^{-1} \in R_p$.
14: **return** $(B, h) \in R^{2N*2N} \times R_p^*$

### A. STRUCTURE OF LB-CLS

In an LB-CLS system, there are seven PPT algorithms, namely, 1) Setup, 2) Partial-Private-Key Extract ($D_{id}$), 3) Set-Secret-Value ($S_{id}$), 4) Set-Private-Key ($SK_{id}$), 5) Set-Public-Key ($PK_{id}$), 6) CL-Sign and 7) CL-Verify, for a user with identity $id$.

- *Setup* $(N)$*:* A security parameter $N$ is utilized as an input by the key generation centre $(KGC)$ to calibrate the private/public key pair $(msk, mpk)$ for himself.
- *Partial-Private-Key Extract* $(msk, id)$*:*In this step, the partial private key $D_{id}$ is calibrated by the $KGC$. Using the $msk$ and identity $id$, the $KGC$ calibrates $D_{id}$ and sends it to user with identity $id$ over a secure channel.
- *Set-Secret-Value* $(id)$*:* The user computes a secret value $S_{(id)}$ as output corresponding to the partial private key $D_{id}$ and identity $id_i$ as inputs.
- *Set-Private-Key* $(D_{id}, S_{id})$*:* The user runs this procedure using $D_{id}$ and $S_{id}$ as inputs to obtain $SK_{id}$ as private key.
- *Set-Public-Key* $(SK_{id})$*:* The user generates the public key $PK_{id}$ for himself by utilizing his private key $SK_{id}$.
- *CL-Sign* $(m, id, SK_{id})$*:* This algorithm generates a signature based on a message $m$, the user's identity $id$, and $SK_{id}$.
- *CL-Verify* $(sig, m, id, PK_{id})$*:* If and only if the input $(sig, m, id, PK_{id})$ is valid, this algorithm returns the signature as *valid* (1). Otherwise, it returns the signature as *invalid* (0).

### B. SECURITY MODEL FOR LB-CLS

A secure LB-CLS scheme must meet the following characteristics:

*Correctness:* The verifier should verify the signature obtained via the CL-Sign generation algorithm.

*Unforgeability:* There are two types of opponents to consider while discussing the LB-CLS system's unforgeability.

- **Type 1**: This kind of adversary is a hostile adversary who can substitute the user's public key with any value he wants.
- **Type 2**: This kind of adversary can access the private key $msk$ of the $KGC$.

The security model is made up of the following two games.

- **Game 1.** This is a game that does interaction between a Type 1 adversary $A_1$ and a challenger $\mathcal{C}$.
- **Game 2.** This is an instance of a game where a Type 2 adversary $A_2$ and a challenger $\mathcal{C}$ interact.

#### 1) GAME 1

The challenger $\mathcal{C}$ and Type 1 adversary $A_1$ are shown interacting in the following game.

*Initialization:* The $\mathcal{C}$ creates the $msk$ using the *Setup* process. As an outside attacker, $A_1$, he is unable to determine the $msk$.

*Queries:* The adversary $A_1$ has the ability to recursively query every oracle, as shown below.

- *Create-User-Oracle:* The oracle maintains the $L_1$-list that contains the 5-tuples $(id, D_{id}, S_{id}, SK_{id}, PK_{id})$.
  The oracle looks up a given identity $id \in (0, 1)^*$ in the $L_1$-list. $PK_{id}$ will be returned as output if the $id$ is found in the $L_1$-list. Otherwise, the oracle uses the *Extract-Partial-Private-Key, Set-Secret-Value, Set-Private-Key,* and *Set-Public-Key* algorithms to generate $id, D_{id}, S_{id}, SK_{id}, PK_{id}$. The oracle then saves $(id, D_{id}, S_{id}, SK_{id}, PK_{id})$ before returning $PK_{id}$.
- *Partial-Private-Key-Extract:* Firstly, the $\mathcal{C}$ examines the $L_1$-list for the identity $id \in (0, 1)^*$. Then, responds with the partial private key to $A_1$.
- *Set-Secret-Value:* During the response challenging game, the secret value $S_{id}$ corresponding with the identity $id$ can be requested by the adversary $A_1$. For the response, the challenger first look in the $L_C$-list and outputs accordingly.Otherwise, produce the secret value for $id$ and sends it to the adversary $A_1$.
- *Replace-Public-Key Challenger:* The adversary $A_1$ can demand to supersede the public key $PK_{id}$ with another value $PK_{id}^*$ associated with the identity. In response to the query, $\mathcal{C}$ revise the list with a new public key.
- *CL-Sign:* With the intake of an $id$, a message $m$, and secret value $S_{id}$ linked with the current public key $PK_{id}$, the $\mathcal{C}$ produces a legitimate signature that can be performed by utilizing the *CL-Sign* algorithm. Note that for the $PK_{id}$ acquired from the *Create-User-Oracle*,$x_{id} = \perp$.

- *Forgery:* The adversary $A_1$ produces a tuple $(id^*, m^*, sig^*, PK_{id})$, where $PK_{id}$ is the original public key. Hence, if the following conditions mentioned below are satisfied, the adversary $A_1$ wins *Game 1*:
  1) For tuple$(id^*, m^*, sig^*, PK_{id})$, the response obtained from the *CL-Verify* algorithm is "accept".
  2) The sign query has never received the request for pair $(id^*, m^*)$.
  3) The *Partial-Private-Key-Extract* query has never received the request corresponding to $id^*$

#### 2) GAME 2

The interaction among the challenger $\mathcal{C}$ and Type 2 adversary $A_2$ in this game is depicted below.

*Initialization:* The *Setup* algorithm is utilizes by the challenger $\mathcal{C}$ and primary secret key *msk* is produced. With the capabilities of the Type 2 adversary $A_2$, he has the knowledge of *msk*.

*Queries:* The following queries are made adaptively by the adversary $A_2$.

- *Create-User-Oracle:* A list of 5-tuples $(id, D_{id}, S_{id}, SK_{id}, PK_{id})$ is maintained by the oracle, which is referred to as the $L_2$-list. The oracle examines the $L_2$-list for an identity $id \in (0,1)^*$. If $id$ is discovered in the $L_2$-list, $\mathcal{C}$ gives $A_2$ the $PK_{id}$. Otherwise, the oracle executes the algorithms: *Set-Secret-Value, Set-Private-Key,* and *Set-Public-Key* in that order to produce $(d_{id}, S_{id}, SK_{id}, PK_{id})$. Lastly, $L_2$-list is updated by the $\mathcal{C}$ as $(id, D_{id}, S_{id}, SK_{id}, PK_{id})$ and $PK_{id}$ is produced as output to the $A_2$.
- *Set-Secret-Value:* $\mathcal{C}$ examines the $L_2$-list for the identity $id \in (0,1)^*$ and gives secret key $S_{id}$ to $A_2$.
- *CL-Sign:* With the intake of an $id$, a message $m$, and secret value $S_{id}$ linked with the current public key $PK_{id}$, the $\mathcal{C}$ produces a legitimate signature that can be performed by utilizing the *CL-Sign* algorithm. Note that for the $PK_{id}$ acquired from the *Create-User-Oracle*,$x_{id} = \perp$.
- *Forgery:* $A_2$ generates a tuple $(id^*, m^*, sig^*, PK_{id})$, where $PK_{id}$ is the original public key being replaced. The winning conditions for the adversary $A_2$ to win Game 2 are as follows:
  1) For tuple$(id^*, m^*, sig^*, PK_{id})$, the response obtained from the *CL-Verify* algorithm is "accept".
  2) The sign query has never received the pair $(id^*, m^*)$.
  3) The *Set-Secret-Value* oracle has never received the $id^*$.

## V. THE PROPOSED LATTICE-BASED CERTIFICATELESS SIGNATURE (LB-CLS) SCHEME

In this section, we first discuss the proposed network model. We then describe different algorithms associated with the proposed scheme.

### A. NETWORK MODEL

As shown in Fig. 2, our proposed model includes important components such as the key generating center (KGC), signer, and verifier. Below is a description of each entity's function.

- **Key Generation Center (KGC):** KGC's responsibility is to create secure communication channels between the signer and the verifier. *KGC* is in charge of creating the system public parameter, partial private key for user and a pair of primary secret key and public key for himself, and for the designed scheme. The *KGC* distributes the remaining keys to the entire network while keeping the primary key to itself.
- **Signer:** An IoT device (PC, smartphone, sensor, etc.) or user can sign documents. The signer generates the signature using his private key.
- **Verifier:** A user or an Internet of Things (IoT) device (PC, automobile, drone, smartphone, sensor, etc.) that has obtained the message from the signer. He needs to examine the authenticity of the received message by utilizing the signer's public key.

**TABLE 2. Notations and their descriptions.**

| Notation | Description |
|---|---|
| $N$ | $N = 2^n > 8$ is a integer |
| $R$ | Set of real numbers |
| $Z$ | Set of integer numbers |
| $Z_p$ | For a prime $p > 0$, the "set of all integers in the interval $(\frac{-p}{2}, \frac{p}{2}]$" |
| $R_p$ | $\frac{Z_p}{(X^N+1)}$, a "polynomial ring with modulo $X^N + 1$ in $Z_p$" |
| $V$ | Mathematical matrix |
| $\|\|V\|\|$ | Mathematical norm of $V$ |
| $D$ | Probability distribution |
| $D_{id}$ | A user's partial private key |
| $H(\cdot)$ | "One way cryptographic hash function" |
| $Sk_{id}$ | A user's private key |
| $S_{id}$ | A user's chosen secret value |
| $Pk_{id}$ | A user's public key |
| $\mathcal{C}$ | Challenger |

### B. CONSTRUCTION OF PROPOSED SIGNATURE SCHEME

Suppose we have a large prime $p = \vec{\Omega}\eta\sqrt{N} \leq 2$ and the security parameter $N$, and $\alpha$, $s$, $\gamma$ are three positive integers such that $s = \Omega((\frac{p}{N})\sqrt{ln(8Np)})$, and $\gamma = 12s\alpha N$. There are two collision-resistant hash functions $H_1 : \{0,1\}^* \rightarrow v \in Z^N$ and $H_2 : Z_p^N \times \{0,1\}^* \rightarrow D_{H^*}\{v \in Z_p^N, 0 \leq ||v||_1 \leq \alpha, \alpha \ll p\}$. The notations used in the proposed scheme are indicated in Table 2.

The proposed scheme is described with the following algorithms:

- *Setup:* The *KGC* accomplishes this phase to create his own primary secret, public key pair $(msk, mpk)$ and system public parameter (Params). This task is listed in Algorithm 4.
- *Partial Private Key:* The *KGC* brings out the following calculations to invoke the partial private keys for a user after obtaining the $id$s and generate the partial private key $D_{id} = (s_1, s_2)$. After getting them, the user can examine the authenticity of the $D_{id}$ by executing Algorithm 5.

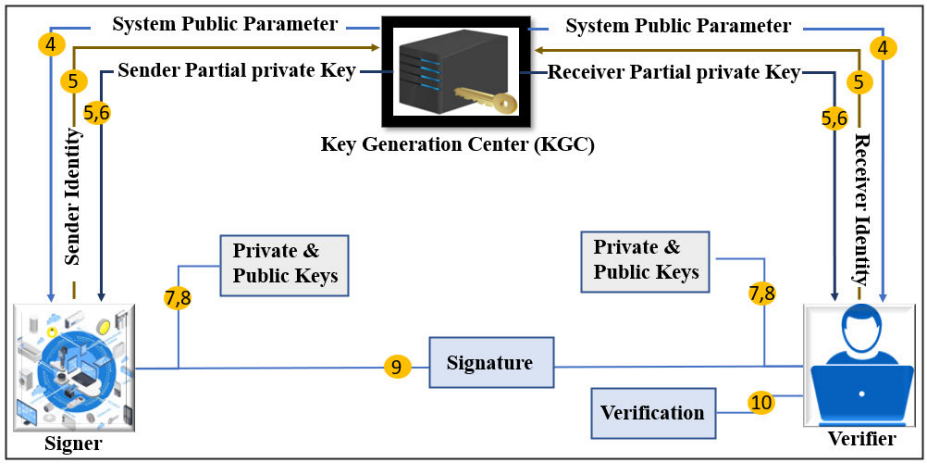


**FIGURE 2.** Proposed network model. (Algorithms 4, 5, 6, 7, 8, 9 and 10 are associated with the proposed scheme.)

**Algorithm 4** Setup ($N$)

**Input:** $N, p \in Z, \gamma > 0$.
**Output:** $(msk, mpk) \in R^{2N\times 2N} \times R_p^*$.

1: Run TrapGen($p, N$).
2: Select $(a, b)$ such that $||a|| \leq s\sqrt{N}$, $||b|| \leq s\sqrt{N}$ and a trapdoor matrix $B = \begin{bmatrix} C(a) & C(b) \\ C(A) & C(B) \end{bmatrix} \in Z_p^{2N\times 2N} = R_p^{2\times 2}$, $h = a^{-1} * b$ of a NTRU lattice $\Lambda_{h,p}$, where $A, B, a, b \in R_q$.
3: Randomly choose $(x_1, x_2) \in Z_p^N$ as $mpk$.
4: Set $msk = B$ and $mpk = (x_1, x_2) \in Z_p^N$.
5: Set Params := $\{(N, s, \gamma, x_1, x_2, H_1, H_2)\}$.
6: **return** $(msk, mpk)$

**Algorithm 5** Partial Private Key Generation ($msk, id$)

**Input:** Hash function $H_1 : (0, 1)^* \rightarrow v \in Z^N$, user identity $id$ and $msk = B$.
**Output:** $D_{id} = (s_1, s_2)$.

1: **if** $D_{id}$ is in disc storage **then**
2: Run PreSampling algorithm on NTRU lattice using GaussSampler($B, s, H_1(id)$,0).
3: Set $(s_1, s_2) \leftarrow [(s, 0) - GaussianSampler(B, s, H_1(id), 0)$ such that $s_1$ and $s_2$ satisfy $[s_1 + s_2 * h = H_1(id], ||(s_1, s_2)|| < s\sqrt{N}$.
4: Set $D_{id} \leftarrow (s_1, s_2)$.
5: Convert $D_{id}$ to a user $id$ and store it on a local disk.
6: **end if**
7: **return** $D_{id}$

**Algorithm 6** Secret Value Generation ($S_{id}$)

**Input:** User's $id$.
**Output:** $S_{id} = (s_3, s_4)$.

1: **if** $S_{id}$ is in disc storage **then**
2: Randomly select $(s_3, s_4) \in D_{Z^N,s}$, $S_{id} \leftarrow (s_3, s_4)$.
3: Convert $S_{id}$ to a user $id$ and store it on a local disk.
4: **end if**
5: **return** $S_{id}$

**Algorithm 7** Private Key Generation ($D_{id}, S_{id}$)

**Input:** User's $id$, $D_{id}$ and $S_{id}$.
**Output:** $Sk_{id} = (D_{id}, S_{id})$.

1: **if** $Sk_{id}$ in disc storage **then**
2: Set $Sk_{id} \leftarrow (D_{id}, S_{id})$.
3: Convert $Sk_{id}$ to a user $id$ and store it on a local disk.
4: **end if**
5: **return** $Sk_{id}$

**Algorithm 8** Public Key Generation ($id, S_{id}$)

**Input:** User's $id$ and secret value $S_{id}$.
**Output:** $Pk_{id} = x_1 * s_3 + x_2 * s_4$.

1: **if** $Pk_{id}$ is in disc storage **then**
2: Set $Pk_{id} \leftarrow (x_1 * s_3 + x_2 * s_4)$.
3: Convert $Pk_{id}$ to a user $id$ and store it on a local disk.
4: **end if**
5: **return** $Pk_{id}$

- *Secret Value:* The user $id$ chooses $s_3, s_4 \in D_{Z^N,s}$ at random and defines $S_{id}$ as his/her secret value. After getting them, the user can examine the authenticity of $S_{id}$ by executing Algorithm 6.

- *Private-Key:* This algorithm defines $Sk_{id} = (D_{id}, S_{id})$ as the private key of the user $id$ and combines the partial private key $D_{id}$ and the secret value $S_{id}$ to form the complete private key $Sk_{id} = (D_{id}, S_{id})$. After getting

**Algorithm 9** Signature Generation $(id, m)$

**Input:** User's identity $id$, message $m$, private key $sk_{id}$ and hash function $H_2 : Z_p^N \times \{0,1\}^* \rightarrow D_{H^*}\{v \in Z_p^N, 0 \leq ||v||_1 \leq \alpha, \alpha \ll p\}$.

**Output:** $Z = (c, z) = (c, z_1, z_2, z_3, z_4)$.

1: Choose randomly $y_1, y_2, y_3, y_4 \in D_{Z^N,s}$.

2: Compute $e = H_2\left(\begin{bmatrix} y_1 + y_2 * h \\ x_1 * y_3 + x_2 * y_4 \end{bmatrix}, m\right)$,

$$z = \begin{bmatrix} z_\alpha \\ z_\beta \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} * e + \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix},$$

where $z_\alpha = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ and $z_\beta = \begin{bmatrix} z_3 \\ z_4 \end{bmatrix}$.

3: Set the signature on $m$ as $Z = (e, z_1, z_2, z_3, z_4)$ with probability of $\min\{\frac{D_{Z^N,s}}{MD_{Z^N,s,Sk_{id}u}}, 1\}$, where $M = O(1)$.

4: **return** $Z$

them, the user can examine the authenticity of $Sk_{id}$ by executing Algorithm 7.

- *Public key:* This algorithm is executed by the user himself, to calibrate his own public key. The algorithm calculates the $PK_{id} = x_1 * s_3 + x_2 * s_4$ as the public key for user. After getting them, the user can examine the authenticity of $Pk_{id}$ by executing Algorithm 8.
- *Signature generation (CL-Sign):* With the inputs $(id, Sk_{id}, H_2)$ and a message $m$, Algorithm 9 is executed to create the signature on $m$.
- *Signature verification (CL-Verify):* After receiving the signed message $(m, id, Pk_{id}, Z) = (e, z_1, z_2, z_3, z_4)$, using the public key $Pk_{id}$, the verification of the signature on the message $m$ is shown as in Algorithm 10.

**Algorithm 10** Signature Verification $(id, m, (e, z_1, z_2, z_3, z_4))$

**Input:** Hash function $H_1$, $H_2$, user's $id, m$, $Z = (e, z_1, z_2, z_3, z_4)$.

**Output:** Valid (1) or Invalid (0).

1: **if** $||z_1|| \leq 2s\sqrt{N}, ||z_2|| \leq 2s\sqrt{N}, ||z_3|| \leq 2s\sqrt{N}, ||z_4|| \leq 2s\sqrt{N}$ **then**

2: Compute $c = H_2\left(\begin{bmatrix} z_1 + z_2 * h \\ b_1 * z_3 + b_2 * z_4 \end{bmatrix} - \begin{bmatrix} H_1(id) \\ Pk_{id} \end{bmatrix} * e, m\right)$.

3: **if** $c = e$ **then**

4: Signature is valid.

5: **return** 1

6: **else**

7: Signature is invalid.

8: **return** 0

9: **end if**

10: **end if**

## VI. SECURITY ANALYSIS

We first provide the correct proof of the proposed signature scheme with respect to the signature verification. Next, we provide a detailed formal security analysis of the proposed scheme.

### A. CORRECTNESS PROOF

The condition $c = e$ should be met for a valid signature. During the CL-Sign phase provided in Algorithm 9 and the CL-Verify phase provided in Algorithm 10 related to the proposed scheme, it follows that.

$$\begin{aligned} &\begin{bmatrix} z_1 + z_2 * h \\ x_1 * z_3 + x_2 * z_4 \end{bmatrix} - \begin{bmatrix} H_1(id) \\ Pk_{id} \end{bmatrix} * e \\ &= \begin{bmatrix} z_1 + z_2 * h \\ x_1 * z_3 + x_2 * z_4 \end{bmatrix} - \begin{bmatrix} s_1 + s_2 * h \\ x_1 * s_3 + x_2 * s_4 \end{bmatrix} * e \\ &= \begin{bmatrix} z_1 - s_1 * e + [z_2 - s_2 * e] * h \\ x_1 * [z_3 - s_3 * e] + x_2 * [z_4 - s_4 * e] \end{bmatrix} \\ &= \begin{bmatrix} y_1 + y_2 * h \\ x_1 * y_3 + x_2 * y_4 \end{bmatrix} \end{aligned}$$

Now,

$$\begin{aligned} c &= H_2\left(\begin{bmatrix} z_1 + z_2 * h \\ x_1 * z_3 + x_2 * z_4 \end{bmatrix} - \begin{bmatrix} H_1(id) \\ Pk_{id} \end{bmatrix} * e, m\right) \\ &= H_2\left(\begin{bmatrix} y_1 + y_2 * h \\ x_1 * y_3 + x_2 * y_4 \end{bmatrix}, m\right) = e. \end{aligned}$$

Furthermore, it is clear that the distributions of $z_1, z_2, z_3$ and $z_4$ are extremely near to $D_{Z^N,s}$. By combining the rejection and Lemma 3, we get $||z_1|| \leq 2s\sqrt{N}$, $||z_2|| \leq 2s\sqrt{N}$, $||z_3|| \leq 2s\sqrt{N}$ and $||z_4|| \leq 2s\sqrt{N}$ with the probability of at least $1 - 2^{-N}$.

### B. SECURITY PROOF

*Theorem 1: Let $N$ be the security parameter and two hash functions, $H_1$ and $H_2$, serve as random oracles. Let our LB-CLS scheme be attacked by a PPT adversary $\mathcal{A}$ (Type 1) with a non-zero probability $\epsilon$. As a result, it is possible to create an algorithm in $\mathcal{C}$ that has a non-zero probability $1 - 2^{w(\log N)}\epsilon$ of solving the SIS problem.*

*Proof:* Let $p$ be prime, $N$ be a positive numeral, and $\gamma, \eta > 0$. Let $\mathcal{C}$ be a challenger who gets a random occurrence of the SIS $(p, 2N, 2k\eta\sqrt{2N} + 4s\sqrt{2N})$. We talk about how the challenger can find a non-zero vector solution $(t_1, t_2) \in R_p^2$ to the SIS problem by interacting with $\mathcal{A}$ (Type 1 adversary), which is what the LB-CLS game says.

*Setup:* The challenger $\mathcal{C}$ controls the random oracles $H_1, H_2$ and chooses polynomials $x_1, x_2, h \in R_q^2$ at random. In the meantime, $\mathcal{C}$ keeps numerous originally empty lists $L_1, L_2, \ldots, L_p$ that are delivered to $\mathcal{A}$.

*Queries:* As shown below, $\mathcal{A}$ can issue many queries to $\mathcal{C}$ in an adaptive manner:

- $H_1$ queries: Let $L_1$ be made up of tuples of the form $[id_j, D_{id_j}, H_1(id_j)]$. When $\mathcal{C}$ receives a query from $\mathcal{A}$ with the $id_j$ parameter, it responds as follows:

In $L_1$, look for $id_j$. Because the query has been issued before, the same answer in $id_j$ is returned to $\mathcal{A}$ if it is found. Otherwise, choose $S_{j1}, S_{j2} \in D_s^N$ at random from the list so that $||S_{j1}, S_{j2}|| < s\sqrt{2N}$ and compute the polynomial $H(id_j) = S_{j1}, S_{j2} * h$. $H(id_j)$ is then sent to $\mathcal{A}$, and $[id_j, D_{id_j} =< S_{j1}, S_{j2} >, H(id_j)]$ is added to the $L_1$ list.

- $H_2$ queries: Let $L_2$ be made up of tuples of the form $[F_j, G_j, m_j, e_j]$. When $\mathcal{C}$ receives a query with $[F_j, G_j, m_j]$ from $\mathcal{A}$, it responds as follows:
  In $L_2$, look for $[F_j, G_j, m_j]$. Because the query has been issued before, the same answer in $L_2$ is provided to $\mathcal{A}$ if it is found. Otherwise, choose $e_j \in Z_p^N$ at random. After that, $e_j$ is transmitted to $\mathcal{A}$, and $[F_j, G_j, m_j, e_j]$ is added to $L_2$'s list.
- Partial private key queries: This query is issued by $\mathcal{A}$ along with $id_j$, and the response from $\mathcal{C}$ is as follows:
  In $L_1$, look for $id_j$. Because the query has been issued before, the same answer in $L_1$ returns to A if it is found. Otherwise, use the $H_1$ query to get the $[id_j, D_{id_j}, H(id_j)]$ tuple return $D_{id_j}$ to $\mathcal{A}$ after that.
- Secret value queries: Let $L_p$ be made up of tuples of the following $[id_j, S_{id_j}, pk_id_j]$. When $\mathcal{C}$ receives a query with $id_j$ from $\mathcal{A}$, it responds in the following manner. In $L_s$, look up $id_j$. Because the query has been issued before, a similar answer in $L_p$ is provided to $\mathcal{A}$ if it is found. Otherwise, pick $S_{j3}, S_{j4} \in (-d, \ldots, 0, \ldots, d)$ where $1 \leq d \leq 31$ and calculate the polynomial $PK_id_j = x_1 * S_{j3} + x_2 * S_{j4}$ at random. $S_{id_j} = s_{j3}, s_{j4}$ is then forwarded to $\mathcal{A}$, and $[id_j, S_{id_j}, pk_id_j]$ are added to the $L_s$ list.
- Public key queries: This query is issued by $\mathcal{A}$ along with $id_j$, and the response from $\mathcal{C}$ is as follows:
  $id_j$ can be found in both $L_1$ and $L_p$. If it is located, $\mathcal{C}$ provides $\mathcal{A}$ with the same answer $PK_{id_j}$, where $PK_{id_j}$ is retrieved from $L_1$ and $L_p$, respectively. Otherwise, use the $H_1$ and Secret Value queries to get the $PK_{id_j}$ values. Then A receives $PK_{id_j}$.
- Public key replacement queries: Challenger $\mathcal{C}$ search for the relevant public key $PK_{id_j}$ and replaces it with $PK'_{id_j}$ given an $id_j$ and a new public key $PK'_{id_j}$. This replacement will finally be recorded.
- Sign queries: When the $\mathcal{C}$ receives a request from $\mathcal{A}$, a message, and $(id_j, PK_{id_j})$, they take the following actions to produce a legitimate signature.
  1) To get $[id_j, D_{id_j}, H_1(id_j)]$, and $[id_j, S_{id_j}, PK_(id_j)]$ search in $L_1$, $L_2$, and $L_p$, respectively.
  2) Arbitrarily choose $Q_j \in (v : v \in (-1, 0, 1 * N, ||v||_1 \leq \alpha$ and $z_1, z_2, z_3, z_4 \in D_s^N$ with $||(z_1, z_2, z_3, z_4)|| \leq 2s\sqrt{4N}$. Then, compute $F_j = z_1 + z_2 * h - H(id_j) * e_j$, $G = x_1 * z_3 + x_2 * z_4 - PK_{id_j} * e_j, m_j$.
  3) Add $F_j, G_j, m_j, e_j$ in the list $L_2$ and send the signature $(z_1, z_2, z_3, z_4, e_j)$ on $m_j$ to $\mathcal{A}$.
  Note that the signature $\phi = (z_1, z_2, z_3, z_4, e_j)$ is valid since it may confirm the above equality: $e_j = H_2(z_1 + z_2 * h - H(id_j) * e_j, x_1 * z_3 + x_2 * z_4 - PK_{id_j} * e_j, m_j) = H_2(F_j, G_j, m_j)$. Thus, regardless of whether challenger $\mathcal{C}$ has the substantial secret key when adversary $\mathcal{A}$ issues the Sign question, challenger $\mathcal{C}$ can deliver a legitimate mark.

*Forgery:* In the wake of playing out the vital questions in general, adversary A makes a mark tuple $(z_1^\bullet, z_2^\bullet, z_3^\bullet, z_4^\bullet, e^\bullet)$ on message $m^\bullet$ for $id^\bullet$. When $\mathcal{A}$ effectively produces a legitimate mark $(z_1^\bullet, z_2^\bullet, z_3^\bullet, z_4^\bullet, e^\bullet)$, $\mathcal{C}$ applies the Forking lemma and replays $\mathcal{A}$ with elective hash upsides of $H_2$ questions to build another substantial mark $(z_1^\star, z_2^\star, z_3^\star, z_4^\star, e^\star)$ on a similar irregular tape as $c^\bullet \neq c\star$. We can get the equality since $(z_1^\bullet, z_2^\bullet, z_3^\bullet, z_4^\bullet, e^\bullet)$ and $(z_1^\star, z_2^\star, z_3^\star, z_4^\star, e^\star)$ are two valid signatures on the message $m^\bullet$ for $(id^\bullet, PK_{id^\bullet})$. We have:

$H_2(z_1^\bullet + z_2^\bullet * h - H_1(id^\bullet) * e_\bullet, x_1 * z_3^\bullet + x_2 * z_4^\bullet - PK_{id^\bullet} * e^\bullet, m^\bullet) = H_2(z_1^\star + z_2^\star * h - H_1(id^\bullet) * e_\star, x_1 * z_3^\star + x_2 * z_4^\star - PK_{id^\bullet} * e^\star, m^\bullet)$, which reduces to $z_1^\bullet + h * z_2^\bullet - H_1(id^\bullet) * e^\bullet = z_1^\star + h * z_2^\star - H_1(id^\bullet) * e^\star$. As a result of $H_1(id^\bullet) = s_1 + s_2 * h$, we arrive at $z_1^\bullet + h * z_2^\bullet - (s_1 + h * s_2) * e^\bullet = z_1^\star + h * z_2^\star - (s_1 + h * s_2) * e^\star$ $z_1^\bullet - z_1^\star - s_1(e^\bullet - e^\star) + h * (z_2^\bullet - z_2^\star - s_2(e^\bullet - e^\star)) = 0$. Thus, $(1, h) * (z_1^\bullet - z_1^\star - s_1(e^\bullet - e^\star), z_2^\bullet - z_2^\star - s_2(e^\bullet - e^\star)) = 0$.

The challenger $\mathcal{C}$ then sets $(t_1, t_2) = z_1^\bullet - z_1^\star - s_1(e^\bullet - e^\star), z_2^\bullet - z_2^\star - s_2(e^\bullet - e^\star)$. We can get $||t_1, t_2|| \leq 2k\eta\sqrt{2N} + 4s\sqrt{4N}$ if $||(z_1^\bullet - z_1^\star, z_2^\bullet - z_2^\star)|| \leq 4s\sqrt{2N}$ and $||(s_1, s_2|| \leq \eta\sqrt{2N}$ have a high probability of being true. As stated, $h = a^{-1} * b$ distribution is statistically close to the uniform distribution of $R_p$. On the NTRU lattice, the SIS problem is to find a pair $(t_1, t_2) \in R_p^2$ such that $t_1 + h * t_2 = 0$ and $||(t_1, t_2)|| \leq 2k\eta\sqrt{2N} + 4s\sqrt{4N}$. We argue that adversary $\mathcal{A}$ solves the SIS problem because he does not know the system secret key B provided by $a, b \in R_q$ and has produced such a pair $(t_1, t_2)$. If the adversary $\mathcal{A}$ is capable of successfully defeating our LB-CLS technique with a non-zero probability $\epsilon$, using the $\mathcal{C}$ we can solve the SIS problem with a non-zero probability: $1 - 2^{-w(\log N)}\epsilon$. ■

*Theorem 2: Let N be the security parameter and two hash functions, $H_1$ and $H_2$, serve as random oracles. Let our LB-CLS system be attacked by a PPT adversary $\mathcal{A}$ (Type 2) with non-zero probability $\epsilon$. The SIS problem can thus be solved with a non-zero probability using the procedure $1 - 2^{-w(\log N)}\epsilon$.*

*Proof:* Let $p$ be a prime number, $N$ be a positive numeral, and $k, \eta, \gamma$ be greater than zero. Let $\mathcal{C}$ be a challenger who is given a random instance of the SIS problem $(p, 2N, 2k\eta\sqrt{2N} + 4s\sqrt{4N})$. We demonstrate how, with the assistance of $\mathcal{A}$, $\mathcal{C}$ can compute a non-zero vector solution $(t_1, t_2)$ to the SIS problem. As described in the LB-CLS's Game-2, $\mathcal{A}$ (Type 2 adversary) interacts with challenger $\mathcal{C}$.

*Setup:* Our LB-CLS scheme's Setup procedure is used to establish $msk = B$ and $Params = (N, k, \eta, s, p, h, H_1,$ and $H_2)$ where two hash functions $H_1$, $H_2$ are random oracles. The *Params* and the secret system key are then transferred to $\mathcal{A}$. $\mathcal{C}$ can compute the partial private key $D_{id}$. Partial public key $H_1(id)$ of every user with $id_j$ without sending any more

queries if he has the system secret key $SK_{id}$. $\mathcal{C}$, on the other hand, keeps several originally empty lists: $L_1$, $L_2$ and $L_p$.

*Queries:* $\mathcal{A}$ can issue many inquiries to $\mathcal{C}$ in an adaptive manner, as follows:

- $H_1$ Queries: $L_1$ should be made up of tuples of the form $[id_j, D_{id_j}, H_1(id_j)]$. When $\mathcal{C}$ receives a query with $id_j$ from $\mathcal{A}$, it responds in the following manner. In $L_1$, look for $id_j$. Because the query has been issued before, the same answer in $L_1$ is returned to $\mathcal{A}$ if it is found. Otherwise, pick a $id_j$ at random and apply the **Gaussian Sampler**$(B,\ \eta, (H_1(id_j),\ 0))$ process to get $s_{j1}, s_{j2} \in D_s^N$ such that $||(s_{1j}, s_{2j})|| < \eta\sqrt{2N}.H_1(id_j)$ is then transmitted to $\mathcal{A}$, and $id_j, D_{id_j}, H_1(id_j)$ is added to $L_1$ 's list.
- $H_2$ Queries: Allow tuples of the form $[F_j, G_j, m_j, e_j]$ to make up $L_2$. When $\mathcal{C}$ receives a query with $[F_j, G_j, m_j]$ from $\mathcal{A}$, it responds as follows:
  In $L_2$, look for $[F_j, G_j, m_j]$. Because the query has been issued before, the same answer in $L_2$ is provided to $\mathcal{A}$ if it is found. Otherwise, choose $e_j \in Z_p^N$ at random. Then $e_j$ is transmitted to $\mathcal{A}$, and $[F_j, G_j, m_j, e_j]$ are added to $L_2$ 's list.
- Secret value queries: Let $L_p$ be made up of tuples of the following form $(id_j, S_{id_j}, PK_{id_j})$. When $\mathcal{C}$ receives a query with $id_j$ from $\mathcal{A}$, it responds in the following manner. In $L_p$, look up $id_j$. Because the query has been issued before, the same answer in $L_p$ is provided to $\mathcal{A}$ if it is found. Otherwise, pick $s_{j3}, s_{j4} \in (-d, \ldots, 0, \ldots d)$, where $1 \le d \le 31$ and solve the polynomial $PK_{id_j} = x_1 * s_{j3} + x_2 * s_{j4}$ at random. Then, $s_{id_j} = (s_{j3}, s_{j4})$ is delivered to $\mathcal{A}$, and $L_p$ is updated with $(id_j, S_{id_j}, PK_{id_j})$.
- Public key queries: This query is issued by $\mathcal{A}$ along with $id_j$, and the response from $\mathcal{C}$ is as follows:
  $id_j$ can be found in both $L_p$. $\mathcal{C}$ provides $\mathcal{A}$ with the same answer $PK_{id_j}$ if it is located. Otherwise, execute the $H_1$ and Secret Value searches to get $PK_{id_j}$. $\mathcal{A}$ receives $PK_{id_j}$.
- Sign queries: After receiving a request from $\mathcal{A}$, a message, and $(id_j, PK_{id_j})$, the challenger $\mathcal{C}$ takes the following actions to create a legitimate signature.
  1) To get $[id_j, D_{id_j}, H_1(id_j)]$, and $[id_j, S_{id_j}, PK_(id_j)]$ search in $L_1$, $L_2$, and $L_p$, respectively.
  2) Randomly choose $Q_j \in (v : v \in (-1, 0, 1 * N, ||v||_1 \le \alpha$ and $z_1, z_2, z_3, z_4 \in D_s^N$ with $||(z_1, z_2, z_3, z_4)|| \le 2s\sqrt{4N}$. Then, compute $F_j = z_1 + z_2 * h - H(id_j) * e_j$, $G = x_1 * z_3 + x_2 * z_4 - PK_{id_j} * e_j, m_j$.
  3) Add $F_j, G_j, m_j, e_j$ in the list $L_2$ and send the signature $(z_1, z_2, z_3, z_4, e_j)$ on $m_j$ to $\mathcal{A}$. Because it may meet the following equality, the signature $\phi = (z_1, z_2, z_3, z_4, e_j)$ is valid: $e_j = H_2(z_1 + z_2 * h - H(id_j) * e_j, x_1 * z_3 + x_2 * z_4 - PK_{id_j} * e_j, m_j) = H_2(F_j, G_j, m_j)$. Therefore, even if challenger $\mathcal{C}$ does not have the correct secret key, the challenger $\mathcal{C}$ can still give a valid signature in response to adversary $\mathcal{A}$'s sign inquiry.

*Forgery:* Subsequent to playing out the essential questions in general, $\mathcal{A}$, makes a mark tuple $(\mathfrak{x}^\bullet, \mathfrak{z}^\bullet, \mathfrak{z}^\bullet, \mathfrak{4}^\bullet, e^\bullet)$ on message $m^\bullet$ for $id^\bullet$. When $\mathcal{A}$ effectively produces a legitimate mark $(z_1^\bullet, z_2^\bullet, z_3^\bullet, z_4^\bullet, e^\bullet)$, the $\mathcal{C}$ applies the Forking lemma and replays $\mathcal{A}$ with elective hash upsides of $H_2$ inquiries to develop another substantial mark $(z_1^\star, z_2^\star, z_3^\star, z_4^\star, e^\star)$ on the same random tape as $c^\bullet \neq c\star$. We can get the equality since $(z_1^\bullet, z_2^\bullet, z_3^\bullet, z_4^\bullet, e^\bullet)$ and $(z_1^\star, z_2^\star, z_3^\star, z_4^\star, e^\star)$ are two valid signatures on the message $m^\bullet$ for $(id^\bullet, PK_{id\bullet})$. We have,

**TABLE 3. System parameters used in implementation.**

| Parameters | Description |
|---|---|
| Key Generation center | Ubuntu 20.04 laptop |
| Programming language | Python 3.11.6 |
| IoT Device | Udoo Bolt V8 |
| $p$ | $14.61\times10^{40}$ |
| $f$ | $2^8$ |
| $N$ | 2 |

$H_2(z_1^\bullet + z_2^\bullet * h - H_1(id^\bullet) * e_\bullet, x_1 * z_3^\bullet + x_2 * z_4^\bullet - PK_{id\bullet} * e^\bullet, m^\bullet) = H_2(z_1^\star + z_2^\star * h - H_1(id^\bullet) * e_\star, x_1 * z_3^\star + x_2 * z_4^\star - PK_{id\bullet} * e^\star, m^\bullet)$, which reduces to $x_1 * z_3^\bullet + x_2 * z_4^\bullet - {}_{id\bullet} * e^\bullet = x_1 * z_3^\star + x_2 * z_4^\star - PK_{id\bullet} * e^\star$. As a result of $PK_{id\bullet} = x_1 * s_3 + x_2 * s_4$, we arrive at

$x_1 * z_3^\bullet + x_2 * z_4^\bullet - (x_1 * s_3 + x_2 * s_4) * e^\bullet = x_1 * z_3^\star + x_2 * z_4^\star - (x_1 * s_3 + x_2 * s_4) * e^\star$, that is, $x_1 * (z_3^\bullet - z_3^\star - s_3(e^\bullet - e^\star)) + x_2 * (z_4^\bullet - z_4^\star - s_4(e^\bullet - e^\star)) = 0$, or, $(x_1, x_2) * (z_3^\bullet - z_3^\star - s_3(e^\bullet - e^\star), z_4^\bullet - z_4^\star - s_4(e^\bullet - e^\star)) = 0$. The challenger $\mathcal{C}$ then sets $(t_1, t_2) = z_3^\bullet - z_3^\star - s_3(e^\bullet - e^\star), z_4^\bullet - z_4^\star - s_4(e^\bullet - e^\star)$.

We can get $||t_1, t_2|| \le 2k\eta\sqrt{2N} + 4s\sqrt{4N}$ if $||(z_3^\bullet - z_3^\star, z_4^\bullet - z_4^\star)|| \le 4s\sqrt{2N}$ and $||(s_3, s_4|| \le \eta\sqrt{2N}$ have a high probability of being true. As stated, $h = a^{-1} * b$ distribution is statistically close to the uniform distribution of $R_p$. On the NTRU lattice, the SIS problem is to find a pair $(t_1, t_2) \in R_p^2$ such that $t_1 + h * t_2 = 0$ and $||(t_1, t_2)|| \le 2k\eta\sqrt{2N} + 4s\sqrt{4N}$. We contend that the system secret key $B$ provided by $a, b \in R_q$ has generated such a pair $(t_1, t_2)$ that the adversary $\mathcal{A}$ to solve the SIS problem. If the adversary $\mathcal{A}$ is capable of successfully defeating our LB-CLS technique with a non-zero probability $\epsilon$, using $\mathcal{C}$, we can solve the SIS problem with a non-zero probability: $1 - 2^{-w(\log N)}\epsilon$. ■

## VII. PERFORMANCE ANALYSIS

The performance of the suggested system in terms of computing cost, communication costs, and safety features in comparison to existing approaches is examined in the section that follows.

### A. COMPUTATIONAL COST

To evaluate the computational cost, a personal computer (HPP with Intel(R) 3.2 GHz processor, the Windows 11 operating system and 512G bytes memory) and Python language is used to implement various primitives. The utilized system parameters to arrive at a reasonable security level is depicted in Table 3. We focus on the computing costs of various phases of proposed protocol. The amount of time needed for different operations in the suggested scheme is shown in Table 4.

**TABLE 4. Execution time for symbol operations.**

| Symbol | Operation | Time ($\mu s$) |
|---|---|---|
| $T_H$ | Time needed to do a general hash in $Z_p^N$ | 48.10 |
| $T_G$ | The amount of time needed to perform a Gaussian sampling operation from $D_{Z^N,s}$ | 11.32 |
| $T_+$ | Time required for an integer matrix addition or a modular vector addition operation | 1.71 |
| $T_*$ | The amount of time needed to multiply an integer matrix or a vector modularly | 108.43 |

**TABLE 5. Comparison of computational costs.**

| Scheme | CL-Sign cost (in $\mu s$) | CL-Verify cost (in $\mu s$) |
|---|---|---|
| Xie *et al.* [4] | $T_H + 4T_G + 6T_+ + 6T_* = 754.22$ | $T_H + 2T_G + 5T_+ + 5T_* = 621.44$ |
| Xu *et al.* [8] | $2T_H + T_G + 5T_+ + 5T_* = 658.22$ | $2T_H + 2T_+ + 4T_* = 533.34$ |
| Lu *et al.* [30] | $3T_H + T_G + 4T_+ + 4T_* = 596.18$ | $1T_H + T_G + 2T_+ + 2T_* = 379.7$ |
| Yang *et al.* [31] | $3T_H + 7T_+ + 5T_* = 915.28$ | $1T_H + 5T_+ + 5T_* = 598.8$ |
| Deng *et al.* [32] | $3T_H + 6T_+ + 6T_* = 1105.14$ | $1T_H + 4T_+ + 4T_* = 588.66$ |
| Dong *et al.* [33] | $2T_H + 2T_G + 2T_+ + 4T_* = 755.98$ | $1T_H + 2T_G + 2T_+ + 2T_* = 379.7$ |
| Proposed | $2T_H + 4T_+ + 4T_* = 536$ | $T_H + 2T_+ + 3T_* = 376.81$ |

For the CL-Sign and CL-Verify phases, the computational costs of the suggested scheme are $2T_H + 4T_+ + 4T_* = 536$ and $1T_H + 2T_+ + 3T_* = 376.81$, respectively. Similarly, the computational costs for the existing schemes of Xie et al. [4], Xu et al. [8], Lu et al. [30], Yang et al. [31], Deng et al. [32], and Dong et al. [33] are demonstrated in Table 5. Based on the execution time for different operations reported in Tables 4 and 5, the computational costs comparison for the CL-Sign and CL-Verify phases are compared and displayed in Figures 3 and 4, respectively. It is worth noting that the proposed mechanism needs lesser computational cost as compared to the existing schemes of Xie et al. [4], Xu et al. [8], Lu et al. [30], Yang et al. [31], Deng et al. [32], and Dong et al. [33].

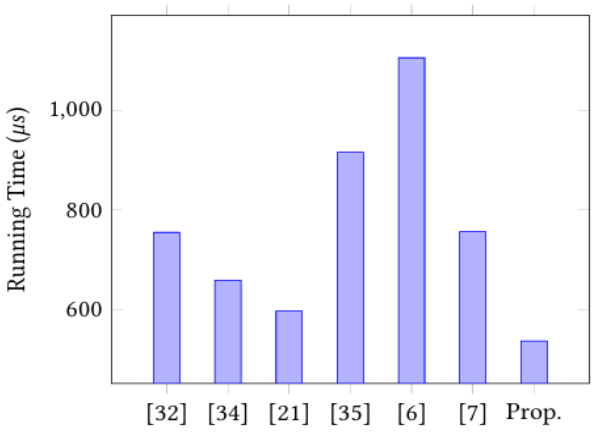


**FIGURE 3. Comparison of computational costs in CL-Sign phase.**

### B. COMMUNICATION COST

Based on our system analysis, the size of the Gaussian function, $Z_p$, polynomial, $p$, and the one-way hash functions are taken as 128, 78, 32, 64, and 20 bits. Therefore, we represented the size of the security parameter at 20 bits, equivalent to the CL-Signer identification and the output of a one-way hash function. We apply the proposed signature scheme in the IoT environment. We have plotted the experimental

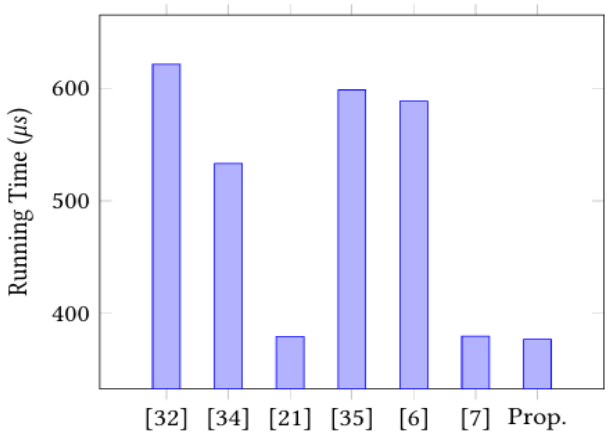


**FIGURE 4. Comparison of computational costs of CL-Verify phase.**

**TABLE 6. Comparison of communication cost (in bits).**

| Scheme | KGC | CL-Sign | CL-Verify |
|---|---|---|---|
| Xie *et al.* [4] | 438 | 552 | 446 |
| Xu *et al.* [8] | 496 | 564 | 484 |
| Lu *et al.* [30] | 478 | 577 | 402 |
| Yang *et al.* [31] | 423 | 492 | 424 |
| Deng *et al.* [32] | 417 | 537 | 397 |
| Dong *et al.* [33] | 446 | 524 | 456 |
| Proposed | 418 | 532 | 446 |

results in Fig.5 which represents a single message signature generation cost on an IoT device. On the other side, Fig. 6 shows a single message signature verification cost on an IoT device for the proposed scheme. It can be observed from figures that the time required for CL-Sign and CL-Verify of single message varies linearly with size of message to signed.

In contrast, the protocols proposed by Xie et al. [4] uses a basic identity based signature mechanism, the Xu et al. [8] use certificateless signature in applied for medical environments, Lu et al. [30], Yang et al. [31], Deng et al. [32], and Dong et al. [33], and the proposed Xu et al. [8] in applied for medical environments. The communication costs corresponding to the KGC, CL-Sign phase, and CL-Verify phase of the proposed mechanism, and other competing

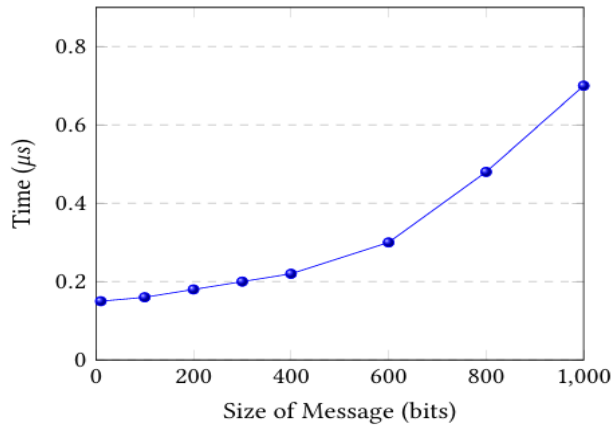


**FIGURE 5.** For the proposed scheme, the cost of single-message signature generation on an IoT device.

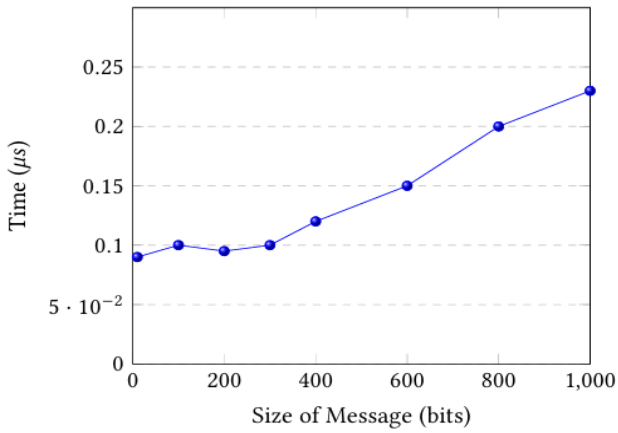


**FIGURE 6.** For the proposed scheme, the cost of single-message signature verification on an IoT device.

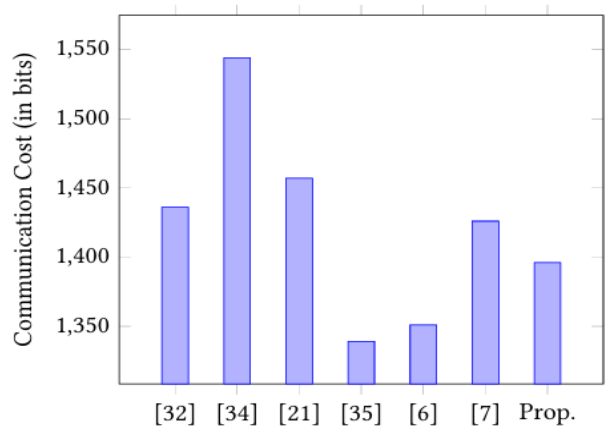


**FIGURE 7.** Comparison of communication costs.

schemes of [4], [8], [30], [31], [32], and [33] are provided in Table 6. The comparison of communication costs between the proposed method and existing schemes in terms of the number of bits utilised is also illustrated in Fig.7. It is pointed out that the proposed scheme's communication costs are considerably higher than those of existing schemes [31], [32] but they have higher computational costs and does not stand against post-quantum security.

### C. SECURITY FEATURES

The security features of the suggested scheme are evaluated with the existing similar protocols in Table 7. The considered parameters for Comparison are the following: a) public key replacement, b) KGC impersonation, c) revocation, d) unforgeability, and e) post-quantum. Lucidly, the proposed mechanism has numerous advantages over existing protocols.

**TABLE 7.** Comparison of capabilities and security features.

| Scheme | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ |
|---|---|---|---|---|---|
| Xie *et al.* [4] | × | ✓ | ✓ | × | ✓ |
| Hung *et al.* [7] | × | ✓ | × | × | ✓ |
| Karati *et al.* [34] | × | × | × | × | × |
| Shim [35] | ✓ | ✓ | × | × | × |
| Xiong *et al.* [36] | ✓ | ✓ | ✓ | ✓ | × |
| Zhang *et al.* [37] | ✓ | ✓ | × | × | × |
| Chen *et al.* [38] | ✓ | ✓ | ✓ | ✓ | × |
| You *et al.* [39] | ✓ | ✓ | ✓ | × | × |
| Lu *et al.* [30] | ✓ | ✓ | × | ✓ | ✓ |
| Yang *et al.* [31] | ✓ | ✓ | × | ✓ | × |
| Deng *et al.* [32] | ✓ | ✓ | ✓ | ✓ | × |
| Dong *et al.* [33] | ✓ | ✓ | ✓ | ✓ | ✓ |
| Proposed | ✓ | ✓ | ✓ | ✓ | ✓ |

**Note:** $A_1$: public key replacement; $A_2$: *KGC* impersonation; $A_3$: revocation; $A_4$: unforgeability; $A_5$: post-quantum security. ×: a scheme is insecure or does not support an attribute; ✓: a scheme is secure or supports an attribute.

**Note:** $A_1$: public key replacement; $A_2$: *KGC* impersonation; $A_3$: revocation; $A_4$: unforgeability; $A_5$: post-quantum security. ×: a scheme is insecure or does not support an attribute; ✓: a scheme is secure or supports an attribute.

## VIII. CONCLUSION

Security is the principal obligatory requirement in an IoT-based network. With the rise of quantum computers, a lattice-based digital signature mechanism is designed, and its applications in IoT are also discussed. The proposed mechanism exploits the benefits of both the certificateless and the lattice-based cryptosystems. Therefore, the proposed mechanism does not face the key escrow problem and can withstand quantum attacks at the same time.Additionally, the suggested scheme's security is robust against adversaries of Type 1 and Type 2. In comparison to existing techniques, the suggested scheme also offers greater security while also providing higher efficiency in terms of lower computational and communication costs. Therefore, the proposed mechanism has better operability and is suitable for IoT-based networks.

In the future we will work on the limitations of the proposed work. We will work on the integration of quantum cryptography with the IoT network and construct a quantum-resistant protocol. We will explore the applications of machine learning and artificial intelligence in IoT based networks.

## REFERENCES

[1] X. Jia, D. He, Q. Liu, and K.-K.-R. Choo, "An efficient provably-secure certificateless signature scheme for Internet-of-Things deployment," *Ad Hoc Netw.*, vol. 71, pp. 78–87, Mar. 2018.

[2] H. Du, Q. Wen, S. Zhang, and M. Gao, "A new provably secure certificateless signature scheme for Internet of Things," *Ad Hoc Netw.*, vol. 100, Apr. 2020, Art. no. 102074.
[3] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 1984, pp. 47–53.
[4] J. Xie, Y.-P. Hu, J.-T. Gao, and W. Gao, "Efficient identity-based signature over NTRU lattice," *Frontiers Inf. Technol. Electron. Eng.*, vol. 17, no. 2, pp. 135–142, Feb. 2016.
[5] G. Wu and R. Huang, "An efficient identity-based forward secure signature scheme from lattices," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2021, pp. 626–631.
[6] M. Tian and L. Huang, "Efficient identity-based signature from lattices," in *Proc. 29th IFIP TC 11 Int. Conf. ICT Syst. Secur. Privacy Protection*, Marrakech, Morocco. Cham, Switzerland: Springer, Jan. 2014, pp. 321–329.
[7] Y.-H. Hung, Y.-M. Tseng, and S.-S. Huang, "Lattice-based revocable certificateless signature," *Symmetry*, vol. 9, no. 10, p. 242, Oct. 2017.
[8] Z. Xu, D. He, P. Vijayakumar, K.-K.-R. Choo, and L. Li, "Efficient NTRU lattice-based certificateless signature scheme for medical cyber-physical systems," *J. Med. Syst.*, vol. 44, no. 5, pp. 1–8, May 2020.
[9] H. Yu and Q. Zhang, "Certificateless threshold signature from lattice," *Digit. Commun. Netw.*, vol. 10, no. 4, pp. 965–972, Aug. 2024.
[10] E. Alkim, N. Bindel, J. Buchmann, Ö. Dagdelen, and P. Schwabe, "Tesla: Tightly-secure efficient signatures from standard lattices," *IACR Cryptol. ePrint Arch.*, vol. 2015, p. 755, Jun. 2015.
[11] D. S. Gupta and G. P. Biswas, "Design of lattice-based ElGamal encryption and signature schemes using SIS problem," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 6, Jun. 2018, Art. no. e3255.
[12] T. Güneysu, V. Lyubashevsky, and T. Pöppelmann, "Practical lattice-based cryptography: A signature scheme for embedded systems," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Cham, Switzerland: Springer, Jan. 2012, pp. 530–547.
[13] E. El Moustaine and M. Laurent, "A lattice based authentication for low-cost RFID," in *Proc. IEEE Int. Conf. RFID-Technologies Appl. (RFID-TA)*, Nov. 2012, pp. 68–73.
[14] A. Abdallah and X. Shen, "Lightweight security and privacy preserving scheme for smart grid customer-side networks," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1064–1074, May 2017.
[15] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, "Lattice signatures and bimodal Gaussians," in *Proc. 33rd Annu. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA. Cham, Switzerland: Springer, Jan. 2013, pp. 40–56.
[16] T. Oder, T. Pöppelmann, and T. Güneysu, "Beyond ECDSA and RSA: Lattice-based digital signatures on constrained devices," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.
[17] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Jan. 2003, pp. 452–473.
[18] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Algorithmic Number Theory*. Berlin, Heidelberg: Springer, 1998, pp. 267–288.
[19] K.-A. Shim, "Security vulnerabilities of four signature schemes from NTRU lattices and pairings," *IEEE Access*, vol. 8, pp. 85019–85026, 2020.
[20] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-dilithium: A lattice-based digital signature scheme," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2018, pp. 238–268, Feb. 2018.
[21] S. McCarthy, J. Howe, N. Smyth, S. Brannigan, and M. O'Neill, "BEARZ attack FALCON: Implementation attacks with countermeasures on the FALCON signature scheme," *Cryptol. ePrint Arch.*, vol. 2019, pp. 61–71, Jan. 2019.
[22] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, May 2008, pp. 197–206.
[23] J. Hoffstein, J. Pipher, J. M. Schanck, J. H. Silverman, and W. Whyte, "Practical signatures from the partial Fourier recovery problem," in *Proc. Int. Conf. Appl. Cryptography Netw. Secur.* Cham, Switzerland: Springer, Jan. 2014, pp. 476–493.
[24] Z. Liu, Y. Han, and X. Yang, "A compressive sensing-based adaptable secure data collection scheme for distributed wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 6, Jun. 2019, Art. no. 155014771985651.
[25] Y. Zhang, X. Lin, and C. Xu, "Blockchain-based secure data provenance for cloud storage," in *Proc. Int. Conf. Inf. Commun. Secur.* Cham, Switzerland: Springer, Jan. 2018, pp. 3–19.
[26] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, Jan. 2012, pp. 738–755.
[27] V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen, "SWIFFT: A modest proposal for FFT hashing," in *Proc. Int. Workshop Fast Softw. Encryption*. Cham, Switzerland: Springer, Jul. 2008, pp. 54–72.
[28] M. Tian and L. Huang, "Certificateless and certificate-based signatures from lattices," *Secur. Commun. Netw.*, vol. 8, no. 8, pp. 1575–1586, May 2015.
[29] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.
[30] X. Lu, W. Yin, Q. Wen, Z. Jin, and W. Li, "A lattice-based unordered aggregate signature scheme based on the intersection method," *IEEE Access*, vol. 6, pp. 33986–33994, 2018.
[31] W. Yang, S. Wang, and Y. Mu, "An enhanced certificateless aggregate signature without pairings for E-healthcare system," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 5000–5008, Mar. 2021.
[32] L. Deng, Y. Yang, and R. Gao, "Certificateless designated verifier anonymous aggregate signature scheme for healthcare wireless sensor networks," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8897–8909, Jun. 2021.
[33] S. Dong, Y. Yao, Y. Zhou, and Y. Yang, "A lattice-based unordered certificateless aggregate signature scheme for cloud medical health monitoring system," *Peer-Peer Netw. Appl.*, vol. 17, no. 1, pp. 284–296, Jan. 2024.
[34] A. Karati, S. H. Islam, and M. Karuppiah, "Provably secure and lightweight certificateless signature scheme for IIoT environments," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3701–3711, Aug. 2018.
[35] K.-A. Shim, "A new certificateless signature scheme provably secure in the standard model," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1421–1430, Jun. 2019.
[36] H. Xiong, Q. Mei, and Y. Zhao, "Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIoT environments," *IEEE Syst. J.*, vol. 14, no. 1, pp. 310–320, Mar. 2020.
[37] Y. Zhang, R. H. Deng, D. Zheng, J. Li, P. Wu, and J. Cao, "Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5099–5108, Sep. 2019.
[38] Y. Chen, D. Zheng, R. Guo, Y. Zhang, and X. Tao, "A blockchain-based revocable certificateless signature scheme for IoT device," *Int. J. Netw. Secur.*, vol. 23, no. 6, pp. 1012–1027, Jun. 2021.
[39] H. Yu and W. Li, "A certificateless signature for multi-source network coding," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102655.

**SUNIL PRAJAPAT** (Student Member, IEEE) received the M.Sc. degree in mathematics from the Central University of Himachal Pradesh, Dharamshala, India, where he is currently pursuing the Ph.D. degree with the Srinivasa Ramanujan Department of Mathematics. He has authored over 30 academic research articles with a focus on information security, privacy preservation and AI/ML. His research interests include quantum cryptography, post-quantum cryptography, coding theory, blockchain technology, and the practical applications of cryptographic primitives in real-world scenarios. He is a fellow of CSIR. He is a dedicated and renowned Reviewer of several prestigious journals, including IEEE, Taylor & Francis, MDPI, *PLOS One*, Elsevier, and Springer.

**DEEPIKA GAUTAM** received the M.Sc. degree in mathematics from the Central University of Himachal Pradesh, Dharamshala, India, where she is currently pursuing the Ph.D. degree. Her research interests include digital signature, authentication, and blockchain technology.

**PANKAJ KUMAR** received the M.Sc. degree from CCS University, Meerut, India, in 2005, and the Ph.D. degree from Galgotias University, in 2020. He has been an Associate Professor with the Srinivasa Ramanujan Department of Mathematics, Central University of Himachal Pradesh, Dharamshala, Himachal Pradesh. He has published over 40 international academic research papers on information security and privacy preservation. His current research interests include cryptography, wireless network security, information theory, and network coding.

**SRINIVAS JANGIRALA** received the Bachelor of Science and Master of Science degrees from Kakatiya University, India, in 2003 and 2008, respectively, the Master of Technology degree from IIT Kharagpur, in 2011, and the Ph.D. degree from the Department of Mathematics, IIT Kharagpur, in 2017. He is currently an Associate Professor with the Jindal Global Business School, O. P. Jindal Global University, Haryana, India. Prior to this, he was a Research Assistant with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology (IIIT), Hyderabad, India. His research interests include blockchain technology and applications, information security, cryptocurrency, and supply chain. He has authored 47 papers in international journals and conferences in his research areas.

**ASHOK KUMAR DAS** (Senior Member, IEEE) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, India. He was a Visiting Research Professor with Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA, USA. He is currently a Full Professor with the Center for Security, Theory and Algorithmic Research, IIIT, Hyderabad, India. He is also an Adjunct Professor with Korea University, Seoul, South Korea. His Google Scholar H-index is 93 and i10-index is 315 with over 26,300 citations. His research interests include cryptography, system and network security, blockchain, security in the Internet of Things (IoT), the Internet of Vehicles (IoV), the Internet of Drones (IoD), smart grids, smart city, cloud/fog computing, intrusion detection, AI/ML security, and post-quantum cryptography. He has authored over 485 papers in international journals and conferences in the above areas, including over 415 reputed journal articles. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He also served as one of the Technical Program Committee Chairs for the first International Congress on Blockchain and Applications (BLOCKCHAIN 2019), Avila, Spain, in June 2019, the International Conference on Applied Soft Computing and Communication Networks (ACN 2020), Chennai, India, in October 2020, the second International Congress on Blockchain and Applications (BLOCKCHAIN 2020), L'Aquila, Italy, in October 2020, and the International Conference on Applied Soft Computing and Communication Networks (ACN 2023), Bengaluru, India, in December 2023. He is/was on the editorial board of IEEE Transactions on Information Forensics and Security, IEEE Systems Journal, *Journal of Network and Computer Applications* (Elsevier), *Computer Communications* (Elsevier), *International Journal of Communication Systems* (Wiley), *Journal of Cloud Computing* (Springer), *Cyber Security and Applications* (Elsevier), *Alexandria Engineering Journal* (Elsevier), *IET Communications*, *KSII Transactions on Internet and Information Systems*, and *International Journal of Communication Systems* (Wiley). He has been listed in the Web of Science (Clarivate[1]) Highly Cited Researcher 2022 and 2023 in recognition of his exceptional research performance.

**BIPLAB SIKDAR** (Fellow, IEEE) received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. From 2001 to 2007, he was an Assistant Professor with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, where he was an Associate Professor, from 2007 to 2013. He is currently a Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, where he is also the Vice Dean of the Faculty of Engineering. His research interests include the IoT and cyber-physical system security, network security, and network performance evaluation. He is a member of Eta Kappa Nu and Tau Beta Pi. He served as an Associate Editor for IEEE Transactions on Communications, from 2007 to 2012, and an Associate Editor for IEEE Transactions on Mobile Computing, from 2014 to 2017.

[1]Trademarked.