Two-Round Identity-Based Proxy Blind Signature Scheme on Lattices

Quanrun Li, Jian Shen, Chao Lin, Zhichao Wang, and Debiao He, *Member, IEEE*

***Abstract*—As Internet technology develops swiftly, the significance of privacy protection is escalating in the realms of e-commerce, e-government and software security. Due to the combination of the benefits of proxy signatures and blind signatures, the proxy blind signature scheme not only distributes the workload across application networks but also effectively safeguards the confidentiality of sensitive information. Additionally, the identity-based proxy blind signature protocol can avoid the problem of heavy certificate management and is widely used in electronic commerce and other scenarios. However, some identity-based proxy blind signature protocols that rely on the large-integer factorization problem and the discrete logarithm problem are unable to withstand from attacks from quantum computers. Furthermore, current lattice-based proxy blind signature protocols offer only heuristic security and require three rounds of information exchange during the signing phase. In this paper, we introduce a new two-round identity-based proxy blind signature scheme based on lattices. This scheme utilizes a zero-knowledge proof protocol on lattices as its core component to develop an interactive two-round signature scheme that is free from security proof vulnerabilities. Additionally, we validate the security of the proposed protocol within the random oracle model and conduct a performance analysis.**

***Index Terms*—Blind signature, interactive signature protocol, lattices, proxy signature, security proof, two round.**

I. Introduction

THE RAPID development of economy and Internet technology has made e-commerce and other electronic services increasingly important. Taking e-commerce as an example, it primarily uses Internet technology and World Wide Web to improve business operational efficiency among trading partners. E-commerce systems [1] can complete the entire process of data and information exchange for transactions electronically, thus achieving paperless and direct transactions throughout the business operation process. Through e-commerce, the relationship between two parties becomes closer, achieving the goal of obtaining maximum benefits with minimal funds. As is widely known, e-commerce not only brings endless business opportunities to merchants but also provides favorable conditions for competition in the market. Moreover, it has an increasingly significant impact on the overall competitiveness of a country. The specific process of one e-commerce system is shown in Fig. 1.

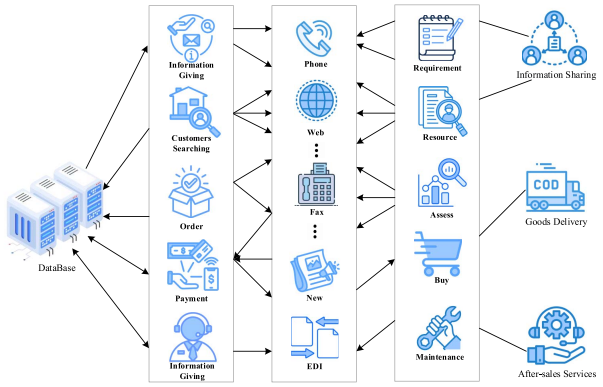

Fig. 1. E-commerce system.

However, e-commerce faces numerous security issues, especially the problem of privacy leakage. As a result, in the contemporary business setting, Internet transactions must integrate security services that ensure authentication and nonrepudiation to prevent fraudulent activities by signatories. To achieve this, digital signatures are introduced into e-commerce. As one of the most important cryptographic primitives, digital signatures ensure data integrity, authentication and nonrepudiation for electronic data. To address the privacy leakage of critical data, blind signature [2] is proposed by Chaum in 1983. They used the blind signature architecture to design an automatic payment system with message privacy protection. This is because blind signatures represent a unique form of digital signature, where a user can obtain a valid signature for any message without revealing any details about that message.

As the number and frequency of transactions increase, more data is stored in the e-commerce system. Consequently, the system center needs to spend considerable resources to generate signatures for all data. To overcome this bottleneck, it is essential for the system center to delegate the signing capability to trusted proxies capable of signing messages on behalf of the center. In 1996, Mambo et al. introduced the concept of proxy signatures [3]. In a proxy signature protocol, the original signer can delegate their signing authority to

Received 17 February 2025; revised 1 April 2025; accepted 28 May 2025. Date of publication 5 June 2025; date of current version 8 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62402443, and in part by the Natural Science Foundation of Zhejiang Province under Grant LQN25F020006. *(Corresponding author: Jian Shen.)*

Quanrun Li and Jian Shen are with the School of Cyber Science and Engineering, Zhejiang Sci-Tech University, Hangzhou 310020, China (e-mail: lqr.ccnu@mails.ccnu.edu.cn; s_shenjian@126.com).

Chao Lin is with the College of Cyber Security, Jinan University, Guangzhou 510610, China (e-mail: linchao91@qq.com).

Zhichao Wang and Debiao He are with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: snkcg@outlook.com; hedebiao@163.com).

Digital Object Identifier 10.1109/JIOT.2025.3576929

another entity, alleviating their workload. As a result, the additional signer, acting as the proxy signer, can sign any message on behalf of the original signer. Meanwhile, the verifier can authenticate and differentiate between the original signature and the proxy signature. Depending on the type of delegation, proxy signatures can be categorized into three types: full delegation [4], partial delegation [5] and delegation by warrant [6].

In modern e-commerce, privacy protection and authority delegation are required simultaneously. In the year 2000, Lin and Jan constructed the first proxy blind signature protocol [7] by integrating the concepts of blind signatures and proxy signatures. In addition, to solve the certificate management problem in the public key infrastructure (PKI) cryptosystem, Shamir proposed the identity (ID)-based cryptosystem in 1985 [8]. The benefit of ID-based cryptosystem is that it eliminates the need to store a large number of certificates, and a user's public key is calculated by the trusted center according to the user's unique identity information. Based on the research of Shamir [8], Zhang and Kim crafted the inaugural ID-based blind signature protocol [9], which combined the benefits of the blind signature framework with the features of ID-based cryptographic systems. Subsequently, Xu et al. integrated proxy signatures into ID-based cryptosystem and constructed an identity-based proxy signature protocol using bilinear pairs [10]. Additionally, in 2004, Dong et al. developed the first identity-based proxy blind signature protocol using elliptic curves [11].

However, quantum computers pose a significant threat to the security of cryptographic schemes that rely on traditional computationally hard problems. To cope with this security problem, four types of post-quantum cryptosystems have been designed, among which lattice-based cryptosystems are considered the most promising for resisting attacks from quantum computers. In 1996, Ajtai [12] proved that the hard mathematical problems on which lattice-based schemes rely cannot be broken by quantum computers. Furthermore, Jiang et al. introduced the initial lattice-based proxy signature protocol in 2010 [13]. Subsequently, Agrawal et al. presented one lattice basis delegation algorithm that preserved the same lattice dimension [14]. In 2012, Cash et al. introduced the critical cryptographic concept known as the bonsai tree [15]. By utilizing lattice basis delegation and bonsai tree techniques, Zhang et al. employed the small integer solution (SIS) problem to devise an identity-based proxy blind signature protocol in 2014 [16]. However, Rawal and Padhye indicated that in Zhang et al.'s lattice-based identity-based proxy blind signature scheme [17], the master secret key could be easily stolen by users. Overall, identity-based proxy blind signature schemes based on lattices may encounter security vulnerabilities or require at least three rounds of information exchange. Consequently, this paper introduces a lattice-based two-round identity-based proxy blind signature scheme.

In Section II, we will discuss the related literature concerning proxy signatures and blind signatures. Section III provides an introduction to the necessary mathematical background on lattices. The network framework, along with security definitions and the security model, is outlined in Section IV. Our proposed two-round proxy blind signature protocol is presented in Section V, with its security proof detailed in Section VI. We assess the performance of the proposed scheme regarding communication costs and computational expenses. Finally, Section VIII offers a summary and conclusion of this article.

II. Review of Literature

Following the initial concept of proxy blind signatures introduced by Lin and Jan [7], which merged the concepts of blind signatures and proxy signatures, Tan et al. [18] developed a proxy blind signature scheme utilizing Schnorr's blind signature framework. However, Awasthi and Lal [19] identified that their scheme was vulnerable to signature forgery attacks and subsequently proposed an enhanced proxy blind signature protocol to address this issue. Sun et al. [20] noted that neither of the previous two proxy blind schemes satisfies unlinkability and did not provide a modification scheme. Additionally, Zhang et al. utilized bilinear pairings to design a proxy blind signature protocol [21]. However, since these proxy blind signature schemes are built upon the PKI cryptosystem, the certificate authority is tasked with managing a vast number of certificates. Due to the challenges associated with certificate management, the operational efficiency of these proxy blind signature schemes is not particularly high.

To circumvent the issue of managing a multitude of certificates, several proxy blind signature schemes have been developed using an identity-based cryptosystem. Building on Shamir's foundational work [8], Sarde and Banerjee [22] employed bilinear pairings to create an ID-based proxy blind signature scheme that concurrently meets the security requirements of both blind and proxy signatures. Tan [23] introduced an efficient identity-based proxy blind signature scheme that eschews the use of bilinear pairings, proving its security in the random oracle model and relying on the discrete logarithm problem.

James et al. [24] introduced an ID-based proxy blind signature protocol that incorporates message recovery, eliminating the need for bilinear pairing computations and making it suitable for devices with limited resources. Furthermore, a variety of other identity-based proxy blind signature schemes have been proposed [25], [26]. These schemes are susceptible to the key escrow problem because the secret keys for all signers are generated by a trusted third party using their identity information, posing a risk to privacy and security.

To eliminate the secuirty issues faced by identity-based proxy blind signature schemes, Meisheng et al. [27] designed the first certificateless proxy blind signature scheme based on the certificateless signature protocol constructed by Li et al. [28]. Furthermore, Song and Zhang [29] improved the certificateless proxy blind signature scheme with bilinear pairings, enhancing its unforgeability and nondeniability.

Liu and You [30] studied the certificateless proxy blind signature protocol within the framework of the PSS standard model, analyzing the formal definition and security model fo certificateless standard signature protocol of PSS standard

model. Besides, they proposed a certificateless proxy blind signature scheme with superior security and high efficiency.

To address the security concerns associated with identity-based proxy blind signature schemes, Meisheng et al. [27] developed the inaugural certificateless proxy blind signature scheme, based on the certificateless signature protocol established by Li et al. [28]. Subsequently, Song and Zhang, [29] enhanced the certificateless proxy blind signature scheme using bilinear pairings, endowing it with robust unforgeability and strong nondeniability. However, He and Mou [31] pointed out that this scheme was unable to resist the public key replacement attack and malicious passive key generation center(KGC), proposing a forgery proxy delegation algorithm. They also improved the original scheme, demonstrating that their enhanced certificateless proxy blind signature scheme addressed the shortcomings of the original scheme. Liu and You [30] conducted a study on the certificateless proxy blind signature protocol within the framework of the PSS standard model, providing a formal definition and security model for the certificateless standard signature protocol. In addition to their analysis, they introduced a certificateless proxy blind signature scheme that boasts enhanced security and high efficiency.

However, quantum computers pose a serious threat to traditional signature protocols, especially with the development of quantum computing [32], [33]. With help of quantum computers, Shor algorithm [34] can easily break the traditional cryptosystems based on discrete logarithms and elliptic curves. In response to this security problem, the National Institute of Standards and Technology (NIST) launched a global solicitation for post-quantum cryptographic algorithm standards in 2016. After four rounds of collection and evaluation, Kyber and HQC were selected as the standards for public encryption algorithms. Meanwhile, Dilithium, Falcon and SPHINCS+ were selected as the standards for signature algorithms. According to the results of these selected algorithms, lattice-based cryptographic protocols are considered the most promising in the quantum era. Using lattice basis delegation technology, Zhang and Sang [35] designed a proxy blind signature scheme. Because this scheme was not constructed based on ideal lattice, its user private key and signature were very large. After that, in 2019, Li et al. [36] proposed a new anti-quantum proxy blind signature scheme for blockchain-enabled Internet of Things (IoT) applications. Their scheme satisfied untraceability and anonymity, thereby enhancing transaction information security on the blockchain-enabled platforms.

Similarly, based on Shamir's identity-based cryptographic architecture [8], a lot of quantum-resistant and identity-based proxy blind signature protocols have been proposed. In 2014, an ID-based proxy-blind signature scheme based on lattices was introduced Zhang and Ma [16], which combined the trapdoor method developed by Gentry et al. [37] and the basis delegation method proposed by Agarwal et al. [14]. Besides, this scheme was proven to be unforgeable and statistically blind. But Rawal and Padhye [17] pointed out that this scheme was insecure by designing an adversary who could reveal the master private key in Zhang et al.'s scheme [16] in 2020. Additionally, in 2023, a lattice-based ID-based proxy blind signature protocol was created by Li et al. [38] and applied to an electronic voting system. However, this scheme has a relatively large signature dimension. To deal with the problem of big signature size in lattice-based proxy blind signature scheme, Yan et al. [39] constructed a novel and efficient ID-based proxy blind signature scheme utilizing lattices. This scheme mainly used an extended algorithm of the left sampling algorithm and the rejection sampling algorithm to generate the proxy blind signature.

Additionally, after the first ID-based proxy signature scheme based on NTRU lattice was designed by Wu et al. [40] and a quantum-resistant ID-based blind signature protocol was introduced by Zhu et al. [41], which took advantage of NTRU lattices, Zhu et al. [41] in 2018 indicated that the protocol of Zhang and Ma [16] was inefficient and proposed an ring-small-integer- solution (RSIS) problem-based proxy blind signature protocol over NTRU lattices. Then, in 2025, Singh et al. [42] demonstrated that Zhu et al.'s NTRU lattices-based signature protocol was not secure. Moreover, they proposed an improved ID-based proxy blind signature scheme using NTRU lattices, and the security of their scheme is based on the approximate shortest vector problem (γ-SVP) and the shortest integer solution problem (SIS). But these schemes either have security issues or require at least three rounds of information exchange.

III. Mathematical Preliminaries

In this section, we introduce some fundamental notions related to identity-based proxy blind signatures. Here, vectors are denoted by lowercase bold letters, such as $\mathbf{a}$. Matrices are denoted by uppercase bold letters, such as $\mathbf{A}$ and its matrix transpose is denoted as $\mathbf{A}^T$. $[n]$ represents the integer set $\{1, 2, \ldots, n\}$. The set of all integers is represented as $\mathbb{Z}$ and the set of all integer coefficient polynomial is represented as $\mathbb{Z}[x]$. The integer coefficient polynomial ring $\mathbb{Z}[x]/(X^n+1)$ is denoted as $\mathbb{R}$ and the polynomial ring $\mathbb{Z}_q[x]/(X^n+1)$ is represented as $\mathbb{R}_q$. χ is a noise distribution defined on the polynomial ring $\mathbb{R}_q$, such as binomial probability distribution. $\| \ \|$ is expressed as the binary norm of a vector or matrix. The symbol gcd represents the greatest common divisor and *span* represents the span space.

A. Lattice

$\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$ are m linearly independent vectors in the n-dimensional Euclidean space. If the matrix $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m)$, one lattice $\Lambda(\mathbf{B})$ generated by $\mathbf{B}$ is a linear combination of integral coefficients of these vectors, defined below

$$\Lambda(\mathbf{B}) = \{\mathbf{Bc} = \sum_{i=1}^{m} \mathbf{b}_i * c_i | \mathbf{c} \in \mathbb{Z}^m\}. \quad (1)$$

In the above equation, c_i is the ith element of $\mathbf{c}$ and all elements of $\mathbf{c}$ are integers. The matrix $\mathbf{B}$ is a basis of the lattice $\Lambda(\mathbf{B})$ and $\tilde{\mathbf{B}}$ is the Schmidt orthogonalization form [43] of the matrix $\mathbf{B}$. Then, two types of q-ary lattices will be introduced.

$$\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{R}_q^m | \mathbf{Ae} = \mathbf{0} \bmod q, \mathbf{A} \in \mathbb{R}_q^{n \times m}\} \quad (2)$$

$$\Lambda^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{R}_q^n | \mathbf{A}^T\mathbf{e} = \mathbf{u} \bmod q, \mathbf{A} \in \mathbb{R}_q^{n \times m}, \mathbf{u} \in \mathbb{R}_q^m\}. \quad (3)$$

It is important to note that the module learning with errors (MLWE) problem [44], the module short integer solution (MSIS) problem [45] and the decisional small matrix ration (DSMR) problem [46] are the hard mathematical problems on which the proposed the proxy blind signature protocol relies. Next, we will describe these problems in detail.

Definition 1 (MSIS Problem): $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are m random and uniformly distributed vectors in $\mathbb{R}_q^d$, the MSIS problem is to find polynomials $z_1, \ldots, z_m \in \mathbb{R}$ satisfying $\sum_{i=1}^{m} \mathbf{a}_i z_i = \mathbf{0} \bmod q$ and $0 < \|\mathbf{z}\| \leq \beta$, where $\mathbf{z} = (z_1, \ldots, z_m)^T \in \mathbb{R}^m$. The module inhomogeneous short integer solution (MISIS) problem involves finding a nonzero polynomial vector $\mathbf{z}$ for a given polynomial vector $\mathbf{y}$ and $\mathbf{z}$ such that $\sum_{i=1}^{m} \mathbf{a}_i z_i = \mathbf{y} \bmod q$ and $0 < \|\mathbf{z}\| \leq \beta$.

Definition 2 (MLWE Problem): Assume that $\mathbf{a} \in \mathbb{R}_q^n$ is a random and uniformly distributed vector, $\mathbf{s} \in \mathbb{R}_q^n$ is a vector with small norm and e is an error polynomial following a discrete Gaussian distribution. MLWE problem is to determine whether a tuple $(\mathbf{a}, u)$ is randomly selected or is equal to $(\mathbf{a}, u = \mathbf{a}\mathbf{s} + e)$.

Definition 3 (DSMR Problem): Assume that four integers $d, k, p, q > 2$ satisfy $\gcd(p, q) = 1$. For an adversary algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ in breaking the DSMR problem is defined as $\mathrm{Adv}^{\mathrm{DSMR}}(\mathcal{A}) = |\Pr[\mathcal{A}(p\mathbf{v}\mathbf{F}^{-1}) \to 1] - \Pr[\mathcal{A}(\mathbf{h}) \to 1]|$, where $(\mathbf{v}, \mathbf{F}) \leftarrow \chi^k \times \chi^{k \times k}$ and the matrix $\mathbf{F}$ module q and p are both invertible, with $\mathbf{h} \leftarrow \mathbb{R}_q^k$. If the advantage $\mathrm{Adv}^{\mathrm{DSMR}}(\mathcal{A})$ is negligible, the DSMR problem is hard.

Because the discrete Gaussian distribution is an important tool in lattice-based cryptosystem, we will give a detailed introduction about it. First, we review the definition of the Gaussian distribution. For any real number $s \geq 0$, the Gaussian distribution with center $\mathbf{c}$ and standard deviation s is defined as

$$\forall \mathbf{x} \in \mathbb{R}^n, \rho_{s,\mathbf{c}}(\mathbf{x}) = e^{\frac{-\pi \|\mathbf{x}-\mathbf{c}\|}{s^2}}. \tag{4}$$

The discrete Gaussian distribution on a lattice Λ is then defined as

$$\forall \mathbf{x} \in \Lambda, \mathcal{D}_{\Lambda,s,\mathbf{c}} = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}. \tag{5}$$

In the following content, a critical lemma about discrete Gaussian distribution and some important algorithms on lattice are described in detail.

Lemma 1: For any lattice Λ, there exists a vector $\mathbf{c} \in Span(\Lambda)$, $\varepsilon \in (0, 1)$ and $s > \eta_\varepsilon(\Lambda)$. And $\eta_\varepsilon(\Lambda)$ is called the smooth parameter, we have the following conclusion:

$$\Pr\left[\mathbf{x} \leftarrow \mathcal{D}_{\Lambda,s,\mathbf{c}} : \|\mathbf{x} - \mathbf{c}\| > s\sqrt{n}\right] \leq \frac{1+\varepsilon}{1-\varepsilon} 2^{-n}. \tag{6}$$

Before introducing three important algorithms in lattice-based cryptosystem, we need to describe a related notion G-Trapdoor [47].

Definition 4 (G-Trapdoor): $\mathbf{a}$ is a random vector in $\mathbb{R}_q^m$ and $\mathbf{g}$ is a gadget vector on the polynomial ring. The G-trapdoor of $\mathbf{a}$ is a matrix $\mathbf{R} \in \mathbb{R}_q^{m \times k}$ that satisfies the equation $\mathbf{a}^T\mathbf{R} = h\mathbf{g}^T \bmod q$, in which h is a label of the matrix $\mathbf{R}$ and is an invertible element in $\mathbb{R}_q$. And h is usually set to 1.

Algorithm 1 GenTrapG: This algorithm provides a method for generating a random vector $\mathbf{a} \in \mathbb{R}_q^m$ and its G-trapdoor.

Algorithm 1: GenTrapG Algorithm

Input: $\mathbf{a}_0 \in \mathbb{R}_q^{m-k}$, $h \in \mathbb{R}_q$, $m, k > 0$
Output: $\mathbf{a} \in \mathbb{R}_q^m$, $\mathbf{R} \in \mathbb{R}_q^{m \times k}$
1) Choosing $\mathbf{R}' \leftarrow \mathcal{D}_s^{(m-k) \times k}$
2) Return $\mathbf{a} = \begin{pmatrix} \mathbf{a}_0 \\ h\mathbf{g} - \mathbf{R}'^T\mathbf{a}_0 \end{pmatrix}$, $\mathbf{R} = \begin{pmatrix} \mathbf{R}' \\ \mathbf{I} \end{pmatrix} \in \mathbb{R}_q^{m \times k}$

Algorithm 2: SampleGP Algorithm

Input: Random Oracle $\mathcal{O}(u, s)$, Gaussian Parameter $s, u \in \mathbb{R}_q$, $\mathbf{a} \in \mathbb{R}_q^m$, $\mathbf{R} \in \mathbb{R}_q^{m \times k}$, $h \in \mathbb{R}_q$
Output: $\mathbf{z} \in \mathbb{R}_q^m$
1) Computing $\mathbf{z}' \leftarrow \mathcal{O}(u, s)$
2) Return $\mathbf{z} = \mathbf{R}h^{-1}\mathbf{z}' \in \mathbb{R}_q^m$

Algorithm 3: ExtLeft Algorithm

Input: $\mathbf{f} \in \mathbb{R}_q^{m+l}$, $\mathbf{R} \in \mathbb{R}_q^{m \times k}$, Gaussian Parameter s
Output: $\mathbf{R}_f \in \mathbb{R}_q^{(m+l) \times k}$
1) Sampling a random matrix $\mathbf{S} \leftarrow \mathcal{D}_s^{l \times k}$
2) Computing $\mathbf{y} = -\mathbf{b}^T\mathbf{S} + h\mathbf{g}^T \in \mathbb{R}_q^k$
3) Running SampleGP($\mathbf{a}$,$\mathbf{R}$,$\mathbf{y}_j$,s)$\to \mathbf{r}'_j j \in [k]$
4) Setting $\mathbf{R}' = (\mathbf{r}'_1, \ldots, \mathbf{r}'_k)$
5) Return $\mathbf{R}_f = \begin{pmatrix} \mathbf{R}' \\ \mathbf{S} \end{pmatrix} \in \mathbb{R}_q^{(m+l) \times k}$

Algorithm 2 (SampleGP): This algorithm provides a general method for efficiently solving the preimage of a MSIS problem related to polynomial vector $\mathbf{a}$ using G-trapdoor. And the output $\mathbf{z}$ of this algorithm can satisfy the equation $\mathbf{a}^T\mathbf{z} = u \bmod q$.

Algorithm 3 (ExtLeft): This algorithm provides a G-trapdoor extension method defined on the polynomial ring. And the output $\mathbf{R}_f$ of this algorithm can satisfy the equation $\mathbf{f}^T\mathbf{R}_f = h\mathbf{g}^T \bmod q$.

It is important to note that the correctness and security of the above three key lattice-based algorithms can be found in the relevant paper [48].

B. Zero-Knowledge Proof System

The noninteractive zero-knowledge (NIZK) proof system is a crucial component of our lattice-based proxy blind signature protocol. Both the prover and the verifier in our zero-knowledge proof system are provided with a common random string *crs*. Below, we describe some fundamental definitions of the NIZK proof system.

Definition 5 (NIZK Proof System): A NIZK proof system π_{NIZK} for the common string *crs* and the relation $\mathcal{R}$ consists of two oracle-calling probabilistic polynomial-time (PPT) algorithms (*Prove*, *Verify*). These notions are described as belows.

1) $Prove^{O}(crs, X, W) \to \pi / \perp$: Given a witness pair $(X, W) \in \mathcal{R}$ and the common string *crs*, the prover outputs either a proof π or the symbol $\perp$ indicating an abort.

2) *Verify*$^{O}(crs, X, \pi) \rightarrow \perp / \top$: The verifier takes as input the common string, a statement X and a proof π, and outputs $\top$ to indicate acceptance or $\perp$ to indicate rejection.

In our NIZK proof system, we use $\mathcal{L}_{\mathcal{R}} = \{X|\exists W, (X, W) \in \mathcal{R}\}$ to represent the language in $\mathcal{R}$. The NIZK proof system in our protocol needs to satisfy four security attributes: correctness, zero-knowledge, single-proof extractability and multiproof extractability. It is important to note that multiproof extractability and single-proof extractability are stronger than soundness [49]. These four security attributes are introduced in the following content.

Definition 6 (Correctness): The NIZK proof system π_{NIZK} is correct if the probability of algorithm *Prove*$^{O}(crs, X, W)$ outputting $\perp$ is at most negl(λ) for arbitray $\lambda \in \mathbb{N}$, $crs \in \{0, 1\}^l$ and $(X, W) \in \mathcal{R}$. And we can make a conclusion as below.

$$\Pr\Big[Prove^{O}(crs, X, W) \rightarrow \pi: \text{Verify}^{O}(crs, X, \pi) \rightarrow \top | \pi \neq \perp\Big] = 1. \quad (7)$$

Definition 7 (Zero-Knowledge): If an NIZK proof system π_{NIZK} is zero-knowledge, there exists a PPT zero-knowledge simulator $\mathbf{Sim} = (sim_0, sim_1)$ being make up of two algorithms sim_0 and sim_1 such that the following conclusion always holds for any PPT adversary $\mathcal{A}$.

$$\text{Adv}^{ZK}_{\pi_{NIZK}}(\mathcal{A}) = |\Pr\Big[\mathcal{A}^{O, Prove}(crs) = 1\Big] - \Pr\Big[\mathcal{A}^{Sim_0, S}(crs) = 1\Big]| = \text{negl}(\lambda). \quad (8)$$

In the above equation, $\mathcal{S}$ and *Prove* are prove oracles. If $(X, W) \notin \mathcal{R}$, both *Prove* and $\mathcal{S}$ output $\perp$ upon receiving the input tuple (X, W). Otherwise, these two oracles return *Prove*$^{O}(crs, X, W)$ or $Sim_1(crs, X)$ respectively. The probability is also determined by the randomness involved in sampling crs. Besides, we assume that the same statement X is queried by the adversary $\mathcal{A}$ to *Prove* or $\mathcal{S}$ at most twice.

For the convenience of introducing the following two definitions, we need an extractor algorithm **Extract**, which is capable of extracting a witeness for any valid statement and proof pair provided by an adversary. There are two specific types of proof of knowledge: multiproof extractability and single-proof extractability.

Definition 8 (Single-Proof Extractability): If there exists a PPT extractor **SingleExtract**, along with a nonnegligible polynomial $p(\lambda)$ and three constants c_1, c_2, e and they meets one restricted condition, one NIZK proof system π_{NIZK} is single-proof extractability. This condition is that for any $crs \in \{0, 1\}^l$, $X \in \mathcal{L}_{\mathcal{R}}$, $Q_H = \text{poly}(\lambda)$ and a PPT adversary $\mathcal{A}$ making at most Q_H random oracle queries with the probability

$$\Pr\Big[\pi \leftarrow \mathcal{A}^{O}(crs, X): \text{Verify}^{O}(crs, X, \pi) = \top\Big] \geq u(\lambda).$$

we can make the following conclusion

$$\Pr\Big[\mathbf{SingleExtract}^{\mathcal{A}}(crs, X, \pi) \rightarrow W: (X, W) \in \mathcal{R}_{\text{gap}}\Big] \geq \frac{1}{p(\lambda) Q_H^e} u(\lambda)^{c_1} - \text{negl}(\lambda) \quad (9)$$

where the runtime of **SingleExtract** is upper bounded by $c_2 * Time(\mathcal{A})$ and we assume one oracle access to $\mathcal{A}$ takes $Time(\mathcal{A})$.

Definition 9 (Multi-proof Extractability): For an NIZK proof system π_{NIZK} to exhibit multiproof extractability, the existence of a PPT oracle simulator $\mathcal{S}_{crs}$ and a PPT extractor **MultiExtract** is required, meeting the following properties.

1) *CRS Indistinguishability:* The following equation is valid for any PPT adversary $\mathcal{A}$:

$$\text{Adv}^{crs}_{\pi_{NIZK}}(\mathcal{A}) = \Big|\Pr\Big[crs \leftarrow \{0, 1\}^l: \mathcal{A}^{O}(crs) = 1\Big] - \Pr\Big[(\tilde{crs}, \tau) \leftarrow \mathcal{S}_{crs}(1^{\lambda}): \mathcal{A}^{O}(\tilde{crs}) = 1\Big]\Big| = \text{negl}(\lambda) \quad (10)$$

2) *Straight-Line Extractability:* With the following probability, for three constants e_1, e_2, c, a $p(\lambda)$, $Q_H = \text{poly}(\lambda)$ and a PPT advarsary $\mathcal{A}$ can make at most Q_H random oracle queries

$$\Pr\Big[(\tilde{crs}, \tau) \leftarrow \mathcal{S}_{crs}(1^{\lambda}), \{(X_i, \pi_i)\}_{i \in [Q_s]} \leftarrow \mathcal{A}^{O}(\tilde{crs}): \forall i \in [Q_s], \text{Verify}^{O}(\tilde{crs}, X_i, \pi_i) = \top\Big] \geq u(\lambda) \quad (11)$$

we have the following probability

$$\Pr[(\tilde{crs}, \tau) \leftarrow \mathcal{S}_{crs}(1^{\lambda}), \{(X_i, \pi_i)\}_{i \in [Q_s]} \leftarrow \mathcal{A}^{O}(\tilde{crs}) \{W_i \leftarrow MultiExtract(1^{\lambda}, Q_H, Q_s, 1/u, \tau, X_i, \pi_i)\}_{i \in [Q_s]} : \forall i \in [Q_s], (X_i, W_i) \in \mathcal{R}_{\text{gap}} \wedge \text{Verify}^{O}(\tilde{crs}, X_i, \pi_i) = \top] \geq \frac{1}{2} u(\lambda) - \text{negl}(\lambda). \quad (12)$$

Additionally, the runtime of *MultiExtract* algorithm is upper bounded by $Q_H^{e_1} Q_s^{e_2} (1/u^c) p(\lambda)$.

IV. System Framework and Security Model

A. System Framework

The proposed lattice-based proxy blind signature protocol involves four types of entities: the trusted third party private key generator (PKG), the central e-commerce server, the edge e-commerce server and the user. Before entering the trading system, all users and e-commerce servers must register with PKG. The PKG is only responsible for all entities' registration and generates their public–private key pairs. The delegation of signing authority occurs between the central e-commmerce server and the edge e-commerce server. A proxy blind signature can be generated after the edge e-commerce server completes information exchange with the target user. Notably, the entire signing phase can be completed without the involvement of the central e-commerce server and PKG, which is a key feature of the system. The system framework, shown in Fig. 2, shows the overall structure.

B. Security Model

It is the proxy blind signature protocol that needs to satisfy blindness and one-more unforgeability. In general, these properties are typically demonstrated through a security game played between a challenger C and a PPT adversary $\mathcal{A}$.

Blindness: Mainly considering the malicious signer, blindness can be proved through a game $\text{Game}_S^{\text{blind}}$ between a

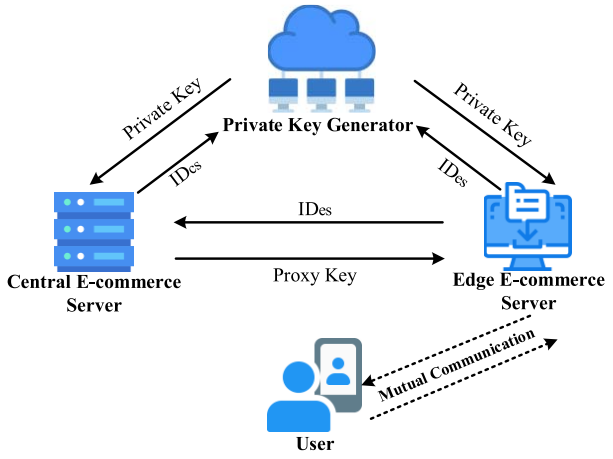

Fig. 2. NetFrame.

malicious signer and two users. Without a nonnegligible probability σ, if no adversary $\mathcal{S}$ can win the game $\text{Game}_{S}^{\text{blind}}$, then the signature protocol can satisfy blindness. Below is how the security game $\text{Game}_{S}^{\text{blind}}$ is described.

$\text{Game}_{S}^{\text{blind}}$: It is in this security game that $\mathcal{U}_1$ and $\mathcal{U}_2$ are two users, and a PPT adversary $\mathcal{S}$ is, the specific process of $\text{Game}_{S}^{\text{blind}}$ played by these entities is as follows.

1) *Setup:* It is a random bit $b \in \{0, 1\}$ that is selected first and this bit is secrect to the adversary $\mathcal{S}$. Randomly chosen by $\mathcal{U}_1$ and $\mathcal{U}_2$ are two messages m_b and m_{1-b}. To $\mathcal{S}$, these two messages are then sent.
2) *Signature:* Once the adversary $\mathcal{S}$ has received the message from $\mathcal{U}_1$ and $\mathcal{U}_2$, it proceeds to execute the blind signature algorithm with $\mathcal{U}_1$ and $\mathcal{U}_2$, utilizing messages m_b and m_{1-b}, respectively. And users $\mathcal{U}_1$ and $\mathcal{U}_2$ generate signatures $\delta(m_b)$ and $\delta(m_{1-b})$. Finally, these two signatures $\delta(m_b)$ and $\delta(m_{1-b})$ are sent to $\mathcal{S}$.
3) *Guess:* It is $\mathcal{S}$ who, after receiving two signatures from $\mathcal{U}_1$ and $\mathcal{U}_2$, needs to guess the value b.

In usual, the advantage of $\mathcal{S}$ who wins the blindness security game is $|\Pr[\text{Game}_{\mathcal{S}}^{Blind} = 1] - (1/2)|$, where $\Pr[\text{Game}_{\mathcal{S}}^{Blind} = 1]$ is the probability that $\text{Game}_{\mathcal{S}}^{Blind} = 1$.

One-More Unforgeability: In this unforgeability security game, it is $\mathcal{F}_1$, a malicious original signer, who knows the proxy key but not the private key of the proxy signer. It is through the security game $\text{Game}_{\mathcal{F}_1}$ that the one-more unforgeability of a proxy-blind signature scheme is demonstrated. If it is not the case that any adversary $\mathcal{F}_1$ wins the game $\text{Game}_{\mathcal{F}_1}$ with nonnegligible probability σ, then the security is established. $\text{Game}_{\mathcal{F}_1}$ is given below.

$\text{Game}_{\mathcal{F}_1}$: $\mathcal{T}$, who acts as the challenger, and $\mathcal{F}_1$, who is the adversary, participate in this game. Additionally, the proxy key is known to the adversary $\mathcal{F}_1$. The game $\text{Game}_{\mathcal{F}_1}$ between the challenger $\mathcal{T}$ and the adversary $\mathcal{F}_1$ is described as follows.

1) *Random Oracle Queries:* If the adversary $\mathcal{F}_1$ were to query the hash value of one message m_i, the challenger $\mathcal{T}$ would return the hash result of m_i to $\mathcal{F}_1$. Furthermore, $\mathcal{F}_1$ would only make random oracle queries in polynomial time.
2) *Signature Queries:* $\mathcal{F}_1$ queries the signature of one message m_i and $\mathcal{T}$ returns the relevant signature ϵ_i to the adversary $\mathcal{F}_1$. After making signature queries to the challenger $\mathcal{T}$, we assume that $\mathcal{F}_1$ got t message-signature pair (m_i, ϵ_i), where $i \in [1, t]$.
3) *Forge:* Only if the signature ϵ^* is valid does the adversary $\mathcal{F}_1$ with the game $\text{Game}_{\mathcal{F}_1}$ in this phase, where $\mathcal{F}_1$ needs to return a forged signature of a message m^*(with $m^* \neq m_i$). Failing otherwise, the advantage of $\mathcal{F}_1$ in winning the game $\text{Game}_{\mathcal{F}_1}$ is determined by the probability of returning a valid signature.

Because our signature protocol combines identity-based cryptosystem with the proxy-blind signature architecture, this protocol also needs to satisfy the following security attributes in addition to blindness and one-more unforgeability.

1) *Distinguishability:* If a signature is generated by the original signer and another by the proxy signer, any user should be able to distinguish between them.
2) *Verifiability:* The verifier can believe that the original signer allows the proxy signer to sign the corresponding message, given that a proxy blind signature is presented.
3) *Strong Identifiability:* If there is only one proxy signature, any user should be able to identify the relevant proxy signer.
4) *Strong Nondeniability:* Not only can a proxy signer generate a valid proxy signature for the original signer, but it cannot deny having generated the signature.

V. Proposed Scheme

The proposed lattice-based proxy blind signature protocol is mainly based on the Commitment-Proof cryptographic architecture, designed by del Pino and Katsumata [50]. Although the zero-knowledge proof protocol on lattices and the lattice-based public key encryption scheme are components of our scheme, they are not the core and are replaceable. Therefore, in this article, we consider them as two functional gadgets to facilitate the overall understanding of our proxy blind signature scheme. Additionally, our scheme also uses a coding function $H : \mathbb{Z}_q^m \to \mathbb{Z}_q^{m \times m}$ designed by Agrawal et al. [14]. For two different inputs $x \in \mathbb{Z}_q^m$ and $y \in \mathbb{Z}_q^m$, $H(x) - H(y) \in \mathbb{Z}_q^{m \times m}$ is invertible and the function value can be computed in polynomial time.

It is the five algorithms (*Setup*, *KeyGen*, *ExtProxyKey*, *Sign* and *Verify*) that our proxy blind signature scheme on lattice consist of, and it is in this section that a detailed description of these five algorithms will be provided.

A. Setup Algorithm

Only after this algorithm is executed by the trusted third party PKG can it output the master private key and master public key for whole system at the end of *Setup*. The specific actions it takes are as follows.

1) PKG uses the system security parameter 1^λ to execute the algorithm GentrapG and generates a polynomial vector $\mathbf{a}_0 \in \mathbb{R}_q^m$ and its trapdoor matrix $\mathbf{R}_0 \in \mathcal{D}_s^{m \times k}$.
2) PKG chooses three random vectors $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{b}$ from $\mathbb{R}_q^m$.
3) Four secure hash functions are selected by PKG, which are $H_1 : \{0, 1\}^* \to \mathbb{R}_q$, $H_2 : \mathbb{Z}_q^m \to \mathbb{R}_q$, $H_3 : \{0, 1\}^* \to \mathbb{R}_q$ and $H_{crs}(\mathbf{0}) = (crs_{NIZK}^M, crs_{com}, \mathbf{a}_3 \in \mathbb{R}_q^2)$. crs_{com}

and crs^M_{NIZK} are common random strings used in multi-proof extractability NIZK and single-proof extractability NIZK.

4) PKG discloses the public system parameter $Param = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{b}, crs_{com}, crs^M_{NIZK}\}$ and the master public key $Mpk = \{\mathbf{a}_0\}$. Meanwhile, PKG keeps the master private key $Msk = \{\mathbf{R}_0\}$ secure.

B. KeyGen Algorithm

This algorithm is executed by PKG as well, which generates private and public keys for the original signer id_1 and the proxy signer id_2. Its operations are as below.

1) PKG uses the following way to computer the public key Upk_{id_1} for signer id_1

$$Upk_{id_1} = \mathbf{f}_{id_1} = \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 + H(id_1)^T\mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{b}_1 \end{bmatrix} \in \mathbb{R}_q^{2m}.$$

2) PKG executes the algorithm $\text{ExLeft}(\mathbf{f}_{id_1}, \mathbf{R}_0, s) \rightarrow \mathbf{R}_{id_1} \in \mathbb{R}_q^{2m\times k}$ that satisfies the equation $\mathbf{f}^T_{id_1}\mathbf{R}_{id_1} = h\mathbf{g}^T \bmod q$.

3) PKG utilizes the following method to generate the public key Upk_{id_2} for signer id_2

$$Upk_{id_2} = \mathbf{f}_{id_2} = \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{a}_2 + H(id_2)^T\mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{b}_2 \end{bmatrix} \in \mathbb{R}_q^{2m}.$$

4) PKG executes the algorithm $\text{ExLeft}(\mathbf{f}_{id_2}, \mathbf{R}_0, s) \rightarrow \mathbf{R}_{id_2} \in \mathbb{R}_q^{2m\times k}$ that satisfies $\mathbf{f}^T_{id_2}\mathbf{R}_{id_2} = h\mathbf{g}^T \bmod q$.

C. ExtProxyKey Algorithm

In this section, a proxy private key for the proxy signer id_2 is generated the original signer id_1. This key indicates that id_1 has authorized id_2 to act as a proxy signer. The specific actions taken by the original signer id_1 are as follows.

1) The signer id_1 generates a proxy public key $Upk_{id_1|id_2}$ through the following method:

$$Upk_{id_1|id_2} = \mathbf{f}_{id_1|id_2} = \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 + H(id_1)^T\mathbf{b} \\ \mathbf{a}_2 + (H(id_1) - H(id_2))^T\mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_3. \end{bmatrix}$$

2) $\text{ExLeft}(\mathbf{f}_{id_1|id_2}, \mathbf{R}_{id_1}, s) \rightarrow \mathbf{R}_{id_1|id_2} \in \mathbb{R}_q^{3m\times k}$ is executed by id_1 and the proxy private key $\mathbf{R}_{id_1|id_2}$ satisfies $\mathbf{f}^T_{id_1|id_2}\mathbf{R}_{id_1|id_2} = h\mathbf{g}^T \bmod q$.

D. Sign Algorithm

This algorithm, executed by the proxy signer id_2 and a user id_U, is an interactive protocol. Because the blind signature process in our scheme is designed based on Commitment-Proof cryptography architecture, there are only two-round message exchange. Here, we use Enc to represent a secure public-key encryption scheme. $Prove^M$ and $Prove^S$, respectively, represent multiproof extractability NIZK protocol and single-proof extractability NIZK protocol. The specific process of this algorithm is shown in the Fig. 3.

First, the user id_U blinds a message M.

1) id_U computes a hash value $\hat{h} = H_3(M)$ and chooses a random value $\text{rand} \leftarrow \mathbb{R}_q^{2m}$.
2) id_U blinds M by computing a commitment $com = COM(crs_{com}, \hat{h}\mathbf{g}, \text{rand})$.
3) id_U executes the multiproof extractability NIZK protocol $Prove^M(crs^M_{NIZK}, com, crs_{com}, \hat{h}, \text{rand}) \rightarrow \pi_M$ to prove that the commitment com has valid form.
4) id_U sends the message $\rho_1 = (com, \pi_M)$.

Second, the proxy signer id_2 generates a valid signature for the commitment com.

1) After receiving ρ_1 from the user id_U, id_2 verifies the validity of π_M. If verification fails, id_2 terminates the signing process. Otherwise, it proceeds to the next step.
2) id_2 executes the algorithm $ParseCOM(com) \rightarrow \mathbf{t}$.
3) id_2 uses the private key $\mathbf{R}_{id_2}$ to execute the algorithm $\text{SampleGP}(\mathbf{f}_{id_2}, \mathbf{R}_{id_2}, H_3(com)) \rightarrow \mathbf{x}_{id_2}$ and the equation $\mathbf{f}^T_{id_2}\mathbf{x}_{id_2} = H_3(com)$ holds.
4) id_2 computes $y_{id} = \prod_{i=1}^{2} H_2(id_i) \in \mathbb{R}_q$ and a hash value $v = H_1(\mathbf{x}_{id_2}, y_{id}) \in \mathbb{R}_q$.
5) id_2 can calculate $\mathbf{f}_{id}$ and execute the algorithm $\text{ExtLeft}(\mathbf{f}_{id}, \mathbf{R}_{id_1|id_2}, s) \rightarrow \mathbf{R}^{4m\times k}_{id}$. We can know the following equation holds.

$$\mathbf{f}_{id} = \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_3 \\ \mathbf{a}_3 + \mathbf{t}. \end{bmatrix}$$

6) id_2 executes the algorithm $\text{SampleGP}(\mathbf{f}_{id}, \mathbf{R}_{id}, v, s) \rightarrow \mathbf{e} \in \mathbb{R}_q^{4m}$ and the equation $\mathbf{f}^T_{id}\mathbf{e} = v = [\mathbf{a}_0^T|\mathbf{b}_1^T|\mathbf{b}_3^T|(\mathbf{a}_3 + \mathbf{t})^T]\mathbf{e}$ is satisfied.
7) id_2, using id_U's public key to generate a ciphertext $CT = Enc(\mathbf{x}_{id_2}||\mathbf{e})$, can provide id_U with the blind signature $\rho_2 = (\mathbf{f}_{id}, CT)$.

Finally, the user id_U outputs a valid signature π_S.

1) After ρ_2 is received, the ciphertext CT is decrypted using the private key of id_U, and the polynomial vector $\mathbf{x}_{id_2}$ can be obtained by id_U. Then id_U verifies whether two equations $\mathbf{f}^T_{id_2}\mathbf{x}_{id_2} = H_3(com)$ and $\mathbf{f}^T_{id}\mathbf{e} = v$ are true. If these equations do not hold true, id_U does not accept the signature ρ_2. Otherwise, the user id_U do next actions.
2) The user id_U divides the vector $\mathbf{e}$ into $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4)^T$.
3) id_U executes two algorithms $ParseCOM(com) \rightarrow \mathbf{t}$ and $ParseRand(\text{rand}) \rightarrow (\mathbf{r}_i)_{i\in[1,2]}$.
4) id_U rewrites the equation $\mathbf{f}^T_{id}\mathbf{e} = v$ into the following equation:

$$\mathbf{f}^T_{id}\mathbf{e} = \left[\mathbf{a}_0^T|\mathbf{a}_3^T + \hat{h}\mathbf{g}^T|\mathbf{b}_1^T|\mathbf{b}_3^T\right]\begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_4 \\ \mathbf{e}_2 + e_{4,1}\mathbf{r}_1 \\ \mathbf{e}_3 + e_{4,2}\mathbf{r}_2 \end{bmatrix} = \left[\mathbf{a}_0^T|\mathbf{a}_3^T + \hat{h}\mathbf{g}^T|\mathbf{b}_1^T|\mathbf{b}_3^T\right]\tilde{\mathbf{e}} = v.$$

5) id_U executes the single-proof extractability NIZK protocol $Prove^S(\mathbf{a}_0, \mathbf{a}_3, \mathbf{b}_1, \mathbf{b}_2, v, \hat{h}, \tilde{\mathbf{e}}) \rightarrow \pi_S$ and it outputs the final signature π_S.

E. Verify Algorithm

This algorithm is executed by any user acting as a verifier. After receiving the signature π_S, the verifier uses the hash

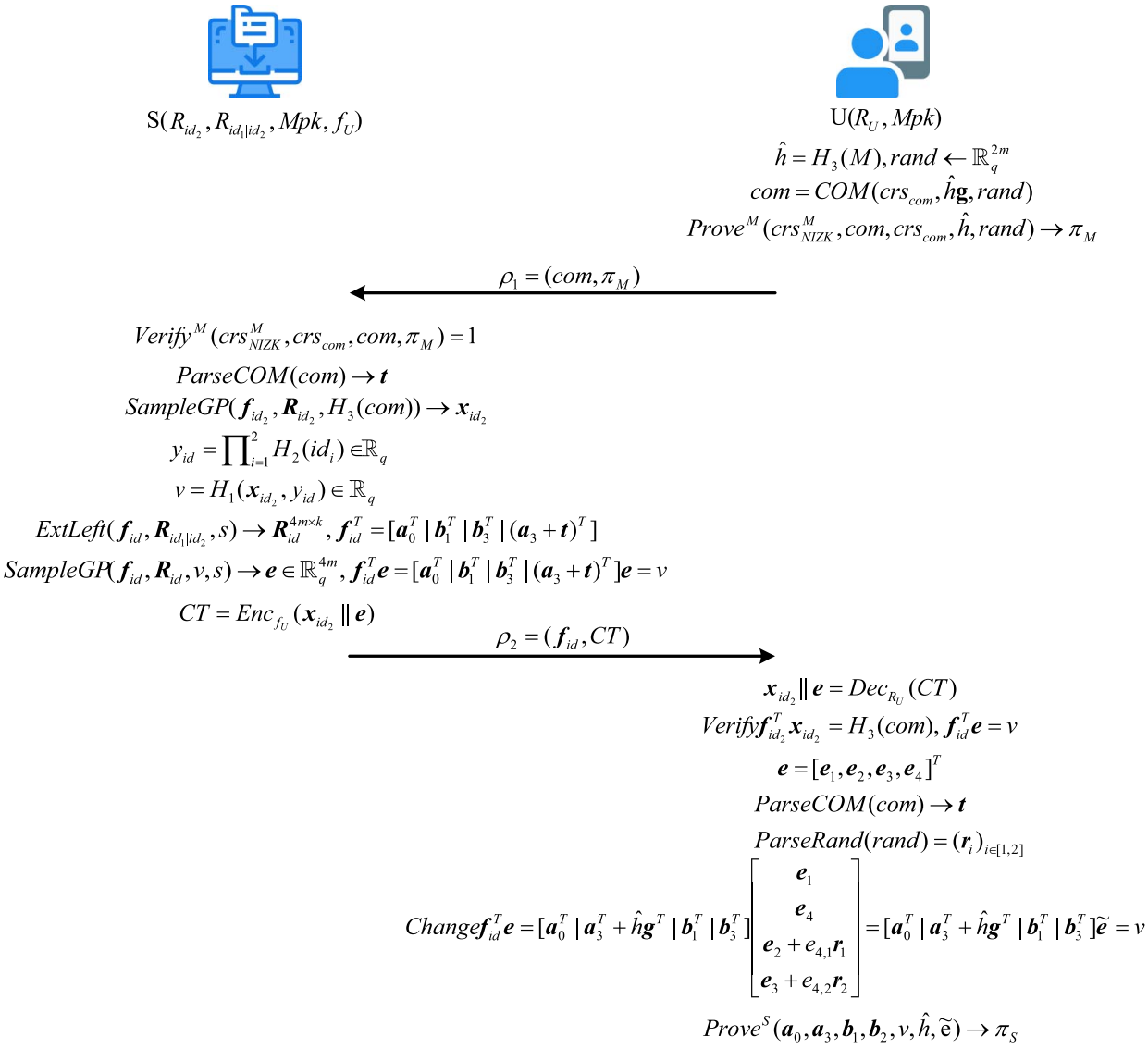

Fig. 3. SignAlgorithm.

value $\hat{h}$ from id_U to verify the validity of this signature. The process is as follows.

1) After obtaining one signature π_S, the verifier can verify its validity, and this is done according to the verification process of the NIZK proof protocol.

Given that the BDLOP commitment protocol *COM* is used in our scheme, it is worth noting that a detailed introduction to *COM* protocol will be provided in the following content. We set $\mathbf{B}_2 = \begin{bmatrix} \mathbf{b}_1^T & 0 \\ 0 & \mathbf{b}_3^T \end{bmatrix}$, the following equation can be computed:

$$\mathbf{t} = \mathbf{B}_2\mathbf{r} + \hat{h}\mathbf{g}^T = \begin{bmatrix} \mathbf{b}_1^T & 0 \\ 0 & \mathbf{b}_3^T \end{bmatrix}\begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} + \hat{h}\begin{bmatrix} g_1 \\ g_2 \end{bmatrix}.$$

We choose random polynomial vectors $\mathbf{b}_0, \mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ and set $\mathbf{B}_1 = \begin{bmatrix} \mathbf{b}_0^T & \mathbf{b}_4^T \\ \mathbf{b}_5^T & \mathbf{b}_6^T \end{bmatrix}$, so the commitment value can be calculated as follows:

$$com = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}\mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \hat{h}\mathbf{g}^T \end{bmatrix} = \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t} \end{bmatrix}.$$

Additionally, to integrate BDLOP commitment well into our proxy blind signature scheme, we need to take $k = 2$ in the whole scheme. Therefore, we can construct the NIZK proof protocol $Prove^M$ based on *COM*, and it is also based on the NIZK protocol designed by del Pino et al. Besides, the protocol $Prove^S$ used in our scheme can be designed by modifying the lattice-based Fiat–Shamir identification scheme. We aim to understand the main idea of our proxy blind signature scheme, and thus we do not expand related descriptions.

Furthermore, provided that the proposed scheme is designed based on "Commitment-Proof" cryptography architecture, the correctness of our proxy blind signature scheme is shown, and the output π_S that NIZK proof protocol $Prove^S$ generates is sent to a verifier as the final signature. The verifier can use the hash value $\hat{h}$ of the message M to verify the validity of π_S if and only if the equation $\mathbf{f}_{id}^T\mathbf{e} = v$ that id_U converted is correct. This equation is converted as follows:

$$\mathbf{f}_{id}^T\mathbf{e} = \left[\mathbf{a}_0^T|\mathbf{b}_1^T|\mathbf{b}_3^T|(\mathbf{a}_3+\mathbf{t})^T\right]\mathbf{e} = \left[\mathbf{a}_0^T|\mathbf{b}_1^T|\mathbf{b}_3^T|(\mathbf{a}_3+\mathbf{t})^T\right]\begin{bmatrix}\mathbf{e}_1\\ \mathbf{e}_2\\ \mathbf{e}_3\\ \mathbf{e}_4\end{bmatrix}$$
$$= \mathbf{a}_0^T\mathbf{e}_1 + \mathbf{b}_1^T\mathbf{e}_2 + \mathbf{b}_3^T\mathbf{e}_3 + (\mathbf{a}_3+\mathbf{t})^T\mathbf{e}_4$$
$$= \mathbf{a}_0^T\mathbf{e}_1 + \mathbf{b}_1^T\mathbf{e}_2 + \mathbf{b}_3^T\mathbf{e}_3 + \mathbf{a}_3^T\mathbf{e}_4 + \left[\mathbf{b}_1^T\mathbf{r}_1 + \hat{h}g_1|\mathbf{b}_2^T\mathbf{r}_2 + \hat{h}g_2\right] * \begin{bmatrix}e_{4,1}\\ e_{4,2}\end{bmatrix}$$
$$= \mathbf{a}_0^T\mathbf{e}_1 + \mathbf{b}_1^T\mathbf{e}_2 + \mathbf{b}_3^T\mathbf{e}_3 + \mathbf{a}_3^T\mathbf{e}_4 + \left(\mathbf{b}_1^T\mathbf{r}_1e_{4,1} + \hat{h}g_1e_{4,1}\right) + \left(\mathbf{b}_3^T\mathbf{r}_2e_{4,2} + \hat{h}g_2e_{4,2}\right)$$
$$= \mathbf{a}_0^T\mathbf{e}_1 + \mathbf{b}_1^T\left(\mathbf{e}_2 + \mathbf{r}_1e_{4,1}\right) + \mathbf{b}_3^T\left(\mathbf{e}_3 + \mathbf{r}_2e_{4,2}\right) + \mathbf{a}_3^T\mathbf{e}_4 + \hat{h}\mathbf{g}^T\mathbf{e}_4$$
$$= \left[\mathbf{a}_0^T|\mathbf{a}_3^T + \hat{h}\mathbf{g}^T|\mathbf{b}_1^T|\mathbf{b}_3^T\right]\begin{bmatrix}\mathbf{e}_1\\ \mathbf{e}_4\\ \mathbf{e}_2 + e_{4,1}\mathbf{r}_1\\ \mathbf{e}_3 + e_{4,2}\mathbf{r}_2\end{bmatrix}$$
$$= \left[\mathbf{a}_0^T|\mathbf{a}_3^T + \hat{h}\mathbf{g}^T|\mathbf{b}_1^T|\mathbf{b}_3^T\right]\tilde{e} = v. \tag{13}$$

VI. Security Proof

In this section, we will formally prove the blindness and one-more unforgeability of the proposed lattice-based proxy blind signature scheme, which satisfies distinguishabillity, verifiability, strong identifiability, strong nondeniability, blindness and one-more unforgeability, and we will also analyze other security attributes.

Theorem 1: With the commitment protocol *COM* being hiding and the two NIZK proof protocol $Prove^M$ and $Prove^S$ being zero-knowledge, the proposed proxy blind signature scheme achieves blindness.

Proof: Let $\mathcal{A}$ be a PPT adversary that attempts to break the blindness of the scheme. We assume that it is against the blindness game and C is a challenger who must respond to valid queries from $\mathcal{A}$. In the following content, we use a series of security games Game_i played by C and $\mathcal{A}$ to prove the above theorem. In every security game Game_i, if the challenger C chooses an initial random value b, the probability ϵ_i that $\mathcal{A}$ outputs the guess value $b' = b$ successfully in this game would be considered. The blindness of our scheme would be proved by showing that differences between ϵ_i in each adjacent games are negligible.

Game₁: This game is the real blindness game that is same as $\text{Game}_S^{\text{blind}}$. In Game_1, the challenger C selects $b = 0$. Meanwhile, $\rho_{1,0}$ and $\rho_{1,1}$ are the first message sent by the challenger C to the adversary $\mathcal{A}$ during two blind signatures are generated. According to the above content, the probability of $\mathcal{A}$ outputting $b' = 0$ is ϵ_1.

Game₂: Different from Game_1, C in this game uses one zero-knowledge simulator $Sim^S = (Sim_0^S, Sim_1^S)$ for $Prove^S$ to respond the NIZK proof query H_S from $\mathcal{A}$ instead of running the NIZK proof protocol $Prove^S$. Concretely, when the adversary $\mathcal{A}$ makes a NIZK proof query H_S, the challenger C runs Sim_0^S. After $\mathcal{A}$ returns the second messages $(\rho_{2,0}, \rho_{2,1})$, the challenger C first extracts $\mathbf{e}_{2,b} \leftarrow \rho_{2,b}, b \in \{0,1\}$. Then C verifies the validity of $\mathbf{e}_{2,0}$ and $\mathbf{e}_{2,1}$. If both $\mathbf{e}_{2,0}$ and $\mathbf{e}_{2,1}$ are valid, C executes $Sim_1^S(\mathbf{a}_0, \mathbf{a}_3, \mathbf{b}_1, \mathbf{b}_2, v, \hat{h}_b) \rightarrow \tilde{\pi}_b^S$, $b \in \{0,1\}$ and $\hat{h}_b = H_3(M_b)$. If $\tilde{\pi}_0^S$ and $\tilde{\pi}_1^S$ are valid, the challenger C uses these two NIZK proofs as the final signatures and sends them to $\mathcal{A}$. In addition, a PPT adversary $\mathcal{B}^S$ can be constructed whose advantage is $|\epsilon_1 - \epsilon_2|$ in the zero-knowledge game. In this game, $\mathcal{B}^S$ can simulate itself as a challenger and interact with $\mathcal{A}$ using Sim_0^S. Meanwhile, $\mathcal{B}^S$ can answer those random oracle queries to H_{crs}, H_3 and $Prove^M$ in an on-the-fly manner. Therefore, we can a conclusion that $|\epsilon_1 - \epsilon_2| \leq \text{Adv}^{Prove^S}(\mathcal{B}^S)$.

Game₃: Instead of running protocol $Prove^M$, this game uses a zero-knowledge simulator $Sim^M = (Sim_0^M, Sim_1^M)$ to respond the queries aiming at NIZK random oracle H_M. Concretely, when a query is made to the oracle H_M by the adversary $\mathcal{A}$, C runs the algorithm Sim_0^M. Moreover, when $\mathcal{A}$ sends two different messages (M_0, M_1) with the same size, C computes $COM(crs_com, \hat{h}_b\mathbf{g}) \rightarrow com_b$, $\hat{h}_b = H_3(M_b)$ and $b \in \{0,1\}$. Then C executes $Sim_1^M(crs_{NIZK}^M, crs_com, com_b) \rightarrow_b^M$ and $b \in \{0,1\}$. At the end, C outputs the first message $\rho_{1,0} = (com_0, \tilde{\pi}_0^M)$ and $\rho_{1,1} = (com_1, \tilde{\pi}_1^M)$ generated by the signing algorithm. Similar to Game_2, a PPT adversary $\mathcal{B}^M$ can be structured such that $|\epsilon_2 - \epsilon_3| \leq \text{Adv}^{Prove^M}(\mathcal{B}^M)$.

Game₄: Different from the game Game_3, the challenger C generates the commitment com_b not by the way of executing $COM(crs_{com}, \hat{h}_b\mathbf{g})$ but by the mean of calculating $COM(crs_{com}, \mathbf{0}) \rightarrow com_b$, and $b \in \{0,1\}$. The output of $H_{crs}(\mathbf{0})$ can be programmed by this game to use crs_{com} provided by *COM*'s hiding game. Therefore, a PPT adversary $\mathcal{B}_{com}$ can be constructed that satisfies the inequality $|\epsilon_3 - \epsilon_4| \leq 2\text{Adv}_{hide}^{COM}(B_{com})$.

According to the above games, we can find that the distribution of messages $(\rho_{1,0}, \rho_{1,1})$ and signatures $(\tilde{\pi}_0^S, \tilde{\pi}_1^S)$ sent to $\mathcal{A}$ are independent of the distribution of the random value b sampled by C. Furthermore, because two NIZK proof protocols and the commitment protocol used in our scheme are secure, $\text{Adv}^{Prove^S}(\mathcal{B}^S)$, $\text{Adv}^{Prove^M}(\mathcal{B}^M)$ and $\text{Adv}_{hide}^{COM}(B_{com})$ are negligible. Therefore, we can make the following conclusion:

$$|\epsilon_1 - \epsilon_4| \leq \text{Adv}^{\text{Prove}^S}\left(\mathcal{B}^S\right) + \text{Adv}^{\text{Prove}^M}\left(\mathcal{B}^M\right) + \text{Adv}_{\text{hide}}^{\text{COM}}(B_{\text{com}}).$$

■

Theorem 2: Provided that MSIS, MLWE, DSMR problems are hard, and the NIZK protocol $Prove^M$ is multiproof extractable and the NIZK protocol $Prove^S$ is single-proof extractable, the proposed proxy blind signature scheme can satisfy one-more unforgeability.

Proof: With $\mathcal{A}$ is a PPT adversary, it can break through one-more unforgeability of our proxy blind signature scheme with a nonnegligible advantage ϵ. Moreover, $\mathcal{A}$ can make Q_S signing queries, Q_M queries to H_3 random oracle, Q_{PM} random queries to $Prove^M$ and Q_{PS} random queries to $Prove^S$. Meanwhile, with the assumption that the adversary $\mathcal{A}$ does not repeat the same query, we will use a series of security games Game_i to prove that the proposed proxy blind signature scheme is one-more unforgeable. Additionally, we use E_i to represent the event that $\mathcal{A}$ wins in Game_i and C in all security games is the challenger.

Game₁: It is this game that played by $\mathcal{A}$ and C in the real world, and it is the one-more unforgeability security game. Besides, it is $\Pr[E_1] = \epsilon$ that represents $\mathcal{A}$'s advantage in this game.

Game$_2$: In this game, C modifies crs^M_{NIZK} of the hash function $H_{crs}(\mathbf{0}) = (crs^M_{NIZK}, crs_{com}, \mathbf{a}_3)$. In Game$_2$, we run one CRS-simulator $S_{crs}(1^\lambda) \to (\tilde{crs}^M_{NIZK}, \tau)$ instead of using a randomly selected value $crs^M_{NIZK} \leftarrow \{0,1\}^*$. Because we want to use $\tilde{crs}^M_{NIZK}$ to meet the requirement of $Prove^M$, we adjust the output crs^M_{NIZK} to $\tilde{crs}^M_{NIZK}$. According to the definition of CRS given in del Pino and Katsumata's paper [50], we can know that Game$_1$ and Game$_2$ are indistinguishable from the adversary $\mathcal{A}$. If to break through the indistinguishability of CRS a PPT adversary $\mathcal{B}_{crs^M_{NIZK}}$ wants, the following conclusion can be made:

$$\Pr[E_2] \geq \Pr[E_1] - \mathrm{Adv}^{crs}_{\pi^M_{NIZK}}\left(\mathcal{B}_{crs^M_{NIZK}}\right).$$

Game$_3$: Different with Game$_2$, the challenger C uses one multiproof extractor *MultiExtract* and those proof in $(\rho_{j,1})_{j\in[Q_S]}$ returned by $\mathcal{A}$ to extract the witness of the relationship R^M_{gap} in this game. The definition of R^M_{gap} can be found in this article [50]. Concretely, when the adversary $\mathcal{A}$ returns $\rho_{j,1} = (com_j, \pi^M_j)$ to C, C runs $MultiExtract(Q_{PM}, Q_S, 1/u, \tau, X_j, \pi^M_j) \to W_j$, where $u = \Pr[E_2]$ and $X_j = (crs_{com}, com_j)$. In addition, we use $Abort_{extract}$ to represent the event that $W_j \notin R^M_{\mathrm{gap}}$ holds for some $j \in [Q_S]$. If the event $Abort_{extract}$ happens, the challenger C aborts in Game$_3$. Otherwise, C can extract the witness W_j. Because C does not make any modifications to $\mathcal{A}$'s interaction, we can make the following conclusion:

$$\Pr[E_3] \geq \frac{1}{2}pr[E_2] - \mathrm{negl}(\lambda).$$

Game$_4$: In this security game, the challenger C needs to guess when the adversary $\mathcal{A}$ will use the target message to be forged for H_3 random oracle query. Provided that C chooses one $j^* \leftarrow [Q_M]$ at the beginning of this game, S_{hash} is set to h_j for $j \in [Q_M]$. After $\mathcal{A}$ sends the jth query message M'_j, C uses h_j to respond this query. In Game$_4$, the challenger C will perform two types of testing: (1) After the challenger C extracts the witeness $W_j = (h'_j, c'_j, c_j, \mathbf{r}_j)$ and the event $Abort_{extract}$ takes place, C checks $h'_j/c'_j \neq h_{j^*}$; (2) Finally, when the forged signature $\{(M_i, \Sigma_i)\}_{i\in[Q_S+1]}$ is returned by $\mathcal{A}$, $M'_{j^*} \in \{M_i\}_{i\in[Q_S+1]}$ and $\{H_3(M_i)\}_{i\in[Q_S+1]}$ are checked by C to see if they are different in pairs. Besides, as long as one of the above two checks fails, we use $Abort_{guess}$ to represent this event. If $Abort_{guess}$ happens, C aborts this security game. Otherwise, we can make the following conclusion:

$$\Pr[E_4] \geq \frac{1}{Q_M}\left(\Pr[E_3] - \frac{Q_M^2+1}{|S_{\mathrm{hash}}|}\right).$$

Game$_5$: Here, C modifies the output $\mathbf{a}_3$ of $H_{crs}(\mathbf{0}) = (crs^M_{NIZK}, crs_{com}, \mathbf{a}_3)$. At the beginning of this security game, C sets $\mathbf{a}_3 = \tilde{\mathbf{a}}_3 - h_j * \mathbf{g}$ and $\tilde{\mathbf{a}}_3 \leftarrow \mathbb{R}^2_q$ after C selects $j^* \leftarrow [Q_M]$ and sets $S_{\mathrm{hash}} \to h_j$, $j \in [Q_M]$. Then C adjusts the output of $H_{crs}(\mathbf{0})$ to a new $\mathbf{a}_3$ instead of $\mathbf{a}_3 \leftarrow \mathbb{R}^2_q$. Because the distribution of these two $\mathbf{a}_3$ is same, we can make a conclusion that $\Pr[E_5] = \Pr[E_4]$.

Game$_6$: In this security game, the challenger C changes the way of generating the blind signature $\mathbf{e}$ after it receives the message ρ_1 sent by the adversary $\mathcal{A}$. Additionally, with the output of the algorithm *BSGen*, C also modifies the output of another algorithm S_2. Those specific operations that C takes are as follows.

$-H_{crs}(\mathbf{0})$: C runs the simulator $S_{crs}(1^\lambda) \to (\tilde{crs}^M_{NIZK}, \tau)$ and chooses a random matrix $\mathbf{R}$. Then C computes $\tilde{\mathbf{a}}_3 = \mathbf{a}_0\mathbf{R}$, where the definition of $\mathbf{a}_0$ is given in *BSGen*. Finally, C adjusts the output of $H_{crs}(\mathbf{0})$ to $(\tilde{crs}^M_{NIZK}, crs_{com}, \mathbf{a}_3 = \tilde{\mathbf{a}}_3 - h_j * \mathbf{g})$.

$-BSGen(1^\lambda)$: The challenger C selects $\mathbf{a}_0 \leftarrow \mathbb{R}^2_q$ and one vector $\mathbf{s}$ with small norm. Meanwhile, C sets $u = [\mathbf{a}_0^T|\mathbf{a}_3^T|\mathbf{b}_1^T|\mathbf{b}_3^T]\mathbf{s} \in \mathbb{R}_q$. Next, C exposes the private and private key pair $(vk, \tilde{sk}) = ((\mathbf{a}_0, u), (\tau, \mathbf{R}))$.

$-S_2(\tilde{sk}, \rho_1)$: C gets $\rho_1 \to (com, \pi^M)$ and it outputs $\perp$ when $\mathrm{Verify}^M(\tilde{crs}^M_{NIZK}, crs_{com}, com, \pi^M) = \perp$. Otherwise, C runs $MultiExtract(1^\lambda, Q^M, Q^S, 1/u, \tau, X, \pi^M) \to W$, where $u = \Pr[E_2]$ and $X = (crs_{com}, com)$. When the event $Abort_{extract}$ does not happen, the equation $W = (h', c', c, (\mathbf{r}_i)_{i\in[2]}) \in R^M_{\mathrm{gap}}$ holds. According to the definition of R^M_{gap} given in del Pino and Katsumata's paper [50], we can get the following equation:

$$\begin{aligned}
&\left[\mathbf{a}_0^T|\mathbf{b}_1^T|\mathbf{b}_2^T|(\mathbf{a}_3+\mathbf{t})^T\right] \\
&= \left[\mathbf{a}_0^T|\mathbf{b}_1^T|\mathbf{b}_2^T|\mathbf{a}_0^T\mathbf{R} - \hat{h}_j\mathbf{g}^T + \left[\frac{\mathbf{b}_1^T\mathbf{r}_1}{c} + \frac{h'}{c'}g_1|\frac{\mathbf{b}_2^T\mathbf{r}_2}{c} + \frac{h'}{c'}g_2\right]\right] \\
&= \left[\mathbf{a}_0^T|\bar{\mathbf{b}}^T|\mathbf{a}_0^T\mathbf{R} + \frac{\bar{\mathbf{b}}^T\hat{\mathbf{R}}}{c} + \left(\frac{h'}{c'} - h^{j^*}\right)\mathbf{g}^T\right] \\
&= \left[\mathbf{a}_0^T|\bar{\mathbf{b}}^T|\left[\mathbf{a}_0^T|\bar{\mathbf{b}}^T\right]\mathbf{R}' + \left(\frac{h'}{c'} - h^{j^*}\right)\mathbf{g}^T\right]\mathbf{P}_{perm}. \qquad (14)
\end{aligned}$$

In the above equation, we can know that $\bar{b}^T = [\mathbf{b}_1^T|\mathbf{b}_2^T]$, $\hat{\mathbf{R}} = \mathbf{I} \otimes [\mathbf{r}_1|\mathbf{r}_2]$ and $\mathbf{R}' = \begin{bmatrix}\mathbf{R} \\ (1/c)\hat{\mathbf{R}}\end{bmatrix}$. In addition, $\mathbf{P}_{perm}$ is a permutation matrix. After that, C runs the algorithm $SampleRight([\mathbf{a}_0^T|\bar{\mathbf{b}}^T], \mathbf{g}, (\mathbf{R}, c, \hat{R}), (h'/c' - h^{j^*}), u, \mathbf{T}_{\mathbf{g}}, \sigma) \to \mathbf{e}'$ to make $\mathbf{e}'$ satisfy the following equation:

$$\left[\mathbf{a}_0^T|\bar{b}^T|[\mathbf{a}_0^T|\bar{b}^T]\mathbf{R}' + \left(\frac{h'}{c'} - h^{j^*}\right)\mathbf{g}^T\right]\mathbf{e}' = u.$$

Because the matrix $\mathbf{P}_{perm}$ is invertible, the challenger C can send $\rho_2 = \mathbf{e}'(\mathbf{P}^{-1}_{perm})$ to the adversary $\mathcal{A}$. The specific description of *SampleRight* can be referred to this article [47] . If there are PPT adversaries $\mathcal{B}_{MLWE}$, $\mathcal{B}'_{\mathrm{DSMR}}$ and $\mathcal{B}_{\mathrm{DSMR}}$ that can break through MLWE problem and DSMR problem, the following conclusion is right:

$$\begin{aligned}
\Pr[E_6] \geq pr[E_5] - \mathrm{Adv}(\mathcal{B}_{MLWE}) - \mathrm{Adv}\left(\mathcal{B}'_{\mathrm{DSMR}}\right) \\
-2\mathrm{Adv}(\mathcal{B}_{\mathrm{DSMR}}) - \mathrm{negl}(\lambda).
\end{aligned}$$

However, due to the fact that MLWE problem and DSMR problem are hard, $\Pr[E_6] \geq \Pr[E_5] - \mathrm{negl}(\lambda)$ is true. According to the above content, if we want to prove that the proposed scheme can satisfy one-more unforgeability, Game$_1$ and Game$_6$ need to be proved to be indistinguishable. Therefore, we need to prove two conclusions in Game$_3$ and Game$_4$. Because proof idea in this two security games is the same as that in del Pino and Katsumata's paper [50], the specific proof details can be referred to their paper. Our scheme can not only satisfy blindness and one-more unforgeability, but also meet distinguishabillity, verifiability, strong identifiability, strong nondeniability.

In the following content, we will make some description about the other four security attributes in detail. ■

Theorem 3: Proposed as an identity-based proxy blind signature scheme, this scheme can satisfy distinguishabillity.

Proof: Because our scheme is constructed by using Commitment-Proof cryptography architecture, the final signature output by this scheme is a publicly verifiable NIZK proof π_S. This signature, which is different from any signature generated by the original signer, ensures that distinguishability is met by our scheme.

Theorem 4: Provided that the identity-based proxy blind signature scheme is proposed, it meets verifiability.

Proof: The verifier can know that the original signer allows the proxy signer to generate a signature for the goal message according to $\mathbf{b}_1$ included in the input of the final NIZK proof protocol and $[\mathbf{a}_0^T|\mathbf{a}_3^T + \hat{h}\mathbf{g}^T|\mathbf{b}_1^T|\mathbf{b}_3^T]\tilde{\mathbf{e}} = v$. Specifically, the input of NIZK proof protocol $Prove^S$ includes $\mathbf{b}_1$ and $\mathbf{b}_1$ is the unique information in original signer id_1's public key, this indicates that the final signature π_S also includes some partial information of id_1. It is most important that the verifier can verify correctness of the equation $[\mathbf{a}_0^T|\mathbf{a}_3^T + \hat{h}\mathbf{g}^T|\mathbf{b}_1^T|\mathbf{b}_3^T]\tilde{\mathbf{e}} = v$. Moreover, because $[\mathbf{a}_0^T|\mathbf{a}_3^T + \hat{h}\mathbf{g}^T|\mathbf{b}_1^T|\mathbf{b}_3^T]\tilde{\mathbf{e}} = v = \mathbf{f}_{id}^T\mathbf{e}$ and the private key $\mathbf{R}_{id}$ corresponding to the public key $\mathbf{f}_{id}^T$ is only generated by the original signer id_1 for the proxy signer id_2, our identity-based proxy blind signature scheme meets verifiability. ■

Theorem 5: The proposed identity-based proxy blind signature scheme satisfies strong identifiability.

Proof: Any user can determine the identity of the proxy signer from the verification process of the final NIZK proof protocol $Prove^S$. Because the input of $Prove^S$ includes the unique information $\mathbf{b}_3$ of the proxy signer, any user can get the corresponding identity information id_2 based on $\mathbf{b}_3$ if and only if π_S is valid. ■

Theorem 6: The proposed identity-based proxy blind signature scheme can meet strong nondeniability.

Proof: When the proxy signer id_2 sends the blind signature $\mathbf{e}$ generated by itself to the verifier, id_2 cannot deny this operation. Specifically, when the verifier receives the message $\rho_2 = (\mathbf{f}_{id}, CT)$ sent by id_2, it can decrypt CT and verify the correctness of the equation $\mathbf{f}_{id_2}^T\mathbf{x}_{id_2} = H_3(com)$. Besides $\mathbf{x}_{id_2}$ can be computed by id_2 using its own private key $\mathbf{R}_{id_2}$. Only when the verifier successfully verifies the previous equation, can it can execute the NIZK proof protocol $Prove^S$ to generate the final signature π_S. ■

VII. Performance Analysis

We make a detailed analysis of the performance of our identity-based proxy blind signature scheme and compare it with other two protocols [38], [39]. All of them are identity-based proxy blind signature protocols on lattices.

To better analyze performance of the proposed identity-based proxy blind signature scheme, we implement this scheme on the hardware platform, which consists of Intel Core i7-10700 CPU @2.9GHz, 8GB RAM and Windows 10 for 64 bit operation system.

TABLE I
Parameter Selections for Our Proxy Blind Scheme

m	k	n	q	s
2	2	256	3329	3

In this part, we first give the parameter selection of the proxy blind signature scheme in this article, and then give the corresponding key and signature size. Because the proposed scheme uses a lattice-based encryption protocol and the NIZK proof protocol on lattice as gadgets, the security levels of these components are relatively independent. Therefore, we can always choose appropriate parameters to make their security level higher than the evaluated security level of the proposed scheme. Due to the fact that the security of the whole scheme in this article depends on the difficulty of the MSIS problem $\mathbf{a}_0^T\mathbf{R}_0 = \mathbf{h}\mathbf{g}^T \bmod q$. In order to achieve the 128 bit security level for the proposed proxy blind signature scheme, the relevant parameters we have selected are shown Table I.

Because our proxy blind scheme is constructed on the identity-based cryptosystem, the evaluation of keys size should be considered from three entities, which are the trusted third party PKG, the original signer and the proxy signer. And the public key of PKG is $Mpk = \mathbf{a}_0$ and its private key is $Msk = \mathbf{R}_0$. Owing to $\mathbf{a}_0 \in \mathbb{R}_q^m$, the size of Mpk is 768 bytes. According to the lemma 1, the maximum size of Msk is 768 bytes. The public key of the original signer id_1 is $Upk_{id_1} = \mathbf{f}_{id_1} \in \mathbb{R}_q^4$, thus the size of Upk_{id_1} is 1536 bytes. The corresponding private key of id_1 is $\mathbf{R}_{id_1} \in \mathbb{R}_q^{4\times 2}$ and the size of this private key is 3072 bytes. It is noting that the proxy signer id_2 needs to save the proxy private key in addition to its own private key. Because the proxy public key that the original signer id_1 generates for the proxy signer id_2 is $Upk_{id_1|id_2} = \mathbf{f}_{id_1|id_2} \in \mathbb{R}_q^6$, the size of $Upk_{id_1|id_2}$ is 2304 bytes. And the size of the corresponding proxy private key $\mathbf{R}_{id_1|id_2}$ is 4608 bytes. Therefore, the size of the public key that the proxy signer id_2 needs to maintain is $1536 + 2304 = 3840$ bytes and the size of the corresponding private key is $3072 + 4608 = 7680$ bytes.

Because the proposed identity-based proxy blind scheme is designed based on the Commitment-Proof cryptography architecture, the final signature that a user outputs is a proof π^S that is generated by the NIZK proof protocol $Prove^S$. The purpose of the user using this NIZK proof protocol $Prove^S$ is to prove the equation $[\mathbf{a}_0^T|\mathbf{a}_3^T + \hat{h}\mathbf{g}^T|\mathbf{b}_1^T|\mathbf{b}_3^T]\tilde{\mathbf{e}} = v$ to any verifier. Additionally, the NIZK proof protocol $Prove^S$ is designed by using the Fiat–Shamir authentication scheme. Its output π^S includes the element $(c, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4)$ in which $c \in \mathbb{R}_q$ and $\mathbf{z}_i \in \mathbb{R}_q^{2\times 1}$. The construction of NIZK proof protocol $Prove^S$ is very simple, the specific description of this protocol can be referred to del Pino and Katsumata's paper [50]. Thus, we can know that the size of the final signature π^S is 3456 bytes.

In the identity-based proxy blind signature scheme proposed by Li et al. [38], the important parameters they have selected are shown Table II. The security level of their lattice-based scheme can also reach 128 bits. The master public key in their scheme is $\mathbf{A} \in \mathbb{Z}_{2q}^{n\times m}$ and the size of this key is 25657344 bytes. Because the master private key $\mathbf{S}$ is generated

TABLE II
PARAMETER SELECTIONS FOR LI ET AL.'S SCHEME

n	m	q	λ	σ_1	σ_2	σ_3
512	13824	2^{27}	128	64	2^{20}	2^{30}

TABLE III
COMPARISON OF KEY SIZE AND SIGNATURE

Protocol	Master Key	Proxy-Private Key	Proxy Blind Signature
Li et al	$mn\log(2q)$	$3mn\log(2q)$	$5m\log(12\sigma)$
Zhou et al	$m^2\log q$	$9m^2\log q$	$5m\log(12\sigma)$
Yan et al	$m^2\log q$	$3mk\log q$	$3m\log(12\sigma)$

by the algorithm $TrapGen(1^{\lambda})$, we can know that $\|\mathbf{S}\| \leq O(\sqrt{n\log 2q})$. According to the fact $\|\mathbf{X}\|_{\infty} \leq \|\mathbf{X}\|$ holds for any vector $\mathbf{X}$, the size of master private key $\mathbf{S}$ is 6193152 bytes.

In their scheme's KeyGen phase, the public key and private key for the original signer ID_o and the proxy signer ID_p are $([\mathbf{A}||H(ID_o)], \mathbf{S}_o)$ and $([\mathbf{A}||H(ID_p)], \mathbf{S}_p)$. Therefore, the size of these two signer's public key are 51314688 bytes. Because $SamplePre(\mathbf{A}||H(ID_o), \mathbf{S}, u, \sigma) \rightarrow \mathbf{S}_o$ and $SamplePre(\mathbf{A}||H(ID_p), \mathbf{S}, u, \sigma) \rightarrow \mathbf{S}_p$, we can know that the size of $\mathbf{S}_o$ and $\mathbf{S}_p$ are 24772608 bytes according to lemma 1. In the ProxyKeyGen phase, the proxy public key $[\mathbf{A}||H(ID_o)||H(ID_p)]$ and the proxy private key $\mathbf{S}'$ can satisfy the equation $[\mathbf{A}||H(ID_o)||H(ID_p)]\mathbf{S}' = q\mathbf{I}_n(\text{mod } 2q)$. Because $[\mathbf{A}||H(ID_o)||H(ID_p)] \in \mathbb{Z}_{2q}^{n\times 3m}$ and $\|\mathbf{S}'\| \leq \sigma\sqrt{3m}$, the size of the proxy public key is 74317824 bytes and the size of the corresponding private key is 37158912 bytes.

In Li et al.'s lattice-based proxy blind signature scheme, the final signature that one user outputs is $(\mathbf{e}_1, \mathbf{e}_2)$, where $\mathbf{e}_1 = \mathbf{y}_1 + \mathbf{z}_1$ and $\mathbf{e}_2 = \mathbf{y}_2 + \mathbf{z}_2$. If the signature $(\mathbf{e}_1, \mathbf{e}_2)$ is valid, we can know that $\|\mathbf{e}_1\|_{\infty} \leq q/4$ and $\|\mathbf{e}_2\|_{\infty} \leq q/4$. Thus, the size of the final signature $(\mathbf{e}_1, \mathbf{e}_2)$ is $86400 + 129600 = 216000$ bytes. In the latest paper on lattice-based proxy blind signature protocols [39], Yan et al. compared their protocol's performance with that of the signature protocol proposed by Li et al. [38] and the protocol created by Zhou et al. [51]. The specific comparison result are show in Table III.

In the above table, these ID-based proxy blind signature protocols used the same parameters to ensure the reliability of the comparison results. Therefore, the comparison of the length of all public–private key pairs among our scheme and other three protocols is shown in Table IV. Moreover, the signature size of these schemes is shorter that of Li et al.'s scheme and the comparison result is shown in Fig. 4. According to the performance analysis and comparison of the above content, our scheme has better communication cost than Li et al.'s scheme [38] and Yan et al.'s scheme [39].

VIII. CONCLUSION

With the development of Internet and mobile communication network technology, proxy blind signature schemes play an increasingly important role to protect sensitive information

TABLE IV
COMPARISON OF PUBLIC–PRIVATE KEY PAIRS (KB)

Description	Our Scheme	Li et al	Yan et al
Master Public Key	0.75	25056	24192
Master Private Key	0.75	6048	23328
Signer's Public Key	1.5	50112	48384
Signer's Private Key	3	24192	93312
Proxy Public Key	2.25	75168	72576
Proxy Private Key	4.5	36288	209952

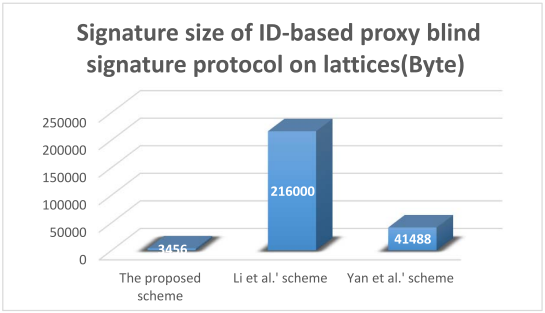

Fig. 4. Signature size of two schemes.

in these networks. In this article, we propose an identity-based proxy blind signature scheme on lattices. This scheme can balance the workload in those network as well. This proxy blind signature only requires two-round information exchange to generate a valid signature with a good size. We proved the blindness and one-more unforgeability of the proposed proxy blind signature scheme and explained how this scheme satisfies other desirable security attributes. Additionally, the result of our scheme's performance analysis demonstrated that this scheme can utilize insignificant communication costs to achieve the goal function and discussed security requirements.

Future research includes designing a certificateless-based proxy blind signature scheme on lattices and improving the efficiency of the corresponding schemes, for example due to changes in other environmental factors.

REFERENCES

[1] S. Burt and L. Sparks, "E-commerce and the retail process: A review," *J. Retail. Consum. Services*, vol. 10, no. 5, pp. 275–286, 2003.
[2] D. Chaum, "Blind signatures for untraceable payments," in *Proc. Adv. Cryptol.*, 1983, pp. 199–203.
[3] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proc. 3rd ACM Conf. Comput. Commun. Security*, 1996, pp. 48–57.
[4] B. Lee, H. Kim, and K. Kim, "Secure mobile agent using strong non-designated proxy signature," in *Proc. 6th Australas. Conf. Inf. Security Privacy (ACISP)*, 2001, pp. 474–486.
[5] T. Okamoto, M. Tada, and E. Okamoto, "Extended proxy signatures for smart cards," in *Proc. 2nd Int. Workshop Inf. Security (ISW)*, 1999, pp. 247–258.
[6] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," in *Proc. 1st Int. Conf. Inf. Commun. Security*, 1997, pp. 223–232.
[7] W. Lin and J. Jan, "A security personal learning tools using a proxy blind signature scheme," in *Proc. Int. Conf. Chin. Lang. Comput.*, 2000, pp. 273–277.
[8] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Adv. Cryptol. (CRYPTO)*, 1985, pp. 47–53.

[9] F. Zhang and K. Kim, "ID-based blind signature and ring signature from pairings," in *Proc. 8th Int. Conf. Theory Appl. Cryptol. Inf. Security Adv. Cryptol. (ASIACRYPT)*, 2002, pp. 533–547.
[10] J. Xu, Z. Zhang, and D. Feng, "ID-based proxy signature using bilinear pairings," in *Proc. Int. Workshops, AEPP, ASTD, BIOS, GCIC, IADS, MASN, SGCA, WISA Parallel Distrib. Process. Appl. ISPA Workshops (ISPA)*, 2005, pp. 359–367.
[11] Z. Dong, H. Zheng, K. Chen, and W. Kou, "ID-based proxy blind signature," in *Proc. 18th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, 2004, pp. 380–383.
[12] M. Ajtai, "Generating hard instances of lattice problems," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 99–108.
[13] Y. Jiang, F. Kong, and X. Ju, "Lattice-based proxy signature," in *Proc. Int. Conf. Comput. Intell. Security*, 2010, pp. 382–385.
[14] S. Agrawal, D. Boneh, and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Proc. 30th Annu. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, 2010, pp. 98–115.
[15] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," *J. Cryptol.*, vol. 25, pp. 601–639, Oct. 2012.
[16] L. Zhang and Y. Ma, "A lattice-based identity-based proxy blind signature scheme in the standard model," *Math. Problems Eng.*, vol. 2014, Sep. 2014, Art. no. 307637.
[17] S. Rawal and S. Padhye, "Cryptanalysis of ID based proxy-blind signature scheme over lattice," *ICT Exp.*, vol. 6, no. 1, pp. 20–22, 2020.
[18] Z. Tan, Z. Liu, and C. Tang, "Digital proxy blind signature schemes based on DLP and ECDLP," *MM Res. Preprints*, vol. 21, no. 7, pp. 212–217, 2002.
[19] A. K. Awasthi and S. Lal, "Proxy blind signature scheme," Cryptol. ePrint Arch., IACR, Bellevue, WA, USA, Rep. 2003/072, 2003.
[20] H.-M. Sun, B.-T. Hsieh, and S.-M. Tseng, "On the security of some proxy blind signature schemes," *J. Syst. Softw.*, vol. 74, no. 3, pp. 297–302, 2005.
[21] F. Zhang, R. Safavi-Naini, and C.-Y. Lin,"Some new proxy signature schemes from pairings," in *Progress on Cryptography: 25 Years of Cryptography China*. Boston, MA, USA: Springer, 2004, pp. 59–66.
[22] P. Sarde and A. Banerjee, "A secure ID-based blind and proxy blind signature scheme from bilinear pairings," *J. Appl. Security Res.*, vol. 12, no. 2, pp. 276–286, 2017.
[23] Z. Tan, "Efficient pairing-free provably secure identity-based proxy blind signature scheme," *Security Commun. Netw.*, vol. 6, no. 5, pp. 593–601, 2013.
[24] S. James, T. Gowri, G. Babu, and P. V. Reddy, "Identity-based blind signature scheme with message recovery," *Int. J. Elect. Comput. Eng.*, vol. 7, no. 5, pp. 2674–2682, 2017.
[25] Q. Wang and Z. Cao, "Identity based proxy multi-signature," *J. Syst. Softw.*, vol. 80, no. 7, pp. 1023–1029, 2007.
[26] X. Bultel, P. Lafourcade, C. Olivier-Anclin, and L. Robert, "Generic construction for identity-based proxy blind signature," in *Proc. 14th Int. Symp. Found. Pract. Security*, 2021, pp. 34–52.
[27] Y. Meisheng, W. Xiaojuan, G. Jian, and Y. Haixia, "A certificateless proxy blind signature scheme," in *Proc. WRI World Congr. Softw. Eng.*, 2009, pp. 177–180.
[28] X. Li, K. Chen, and L. Sun, "Certificateless signature and proxy signature schemes from bilinear pairings," *Lith. Math. J.*, vol. 45, pp. 76–83, Jan. 2005.
[29] M. Song and Y. Zhang, "An improved certificateless proxy blind signature scheme," in *Proc. IEEE 14th Int. Conf. Commun. Technol.*, 2012, pp. 645–649.
[30] L. Liu and Y. You, "Certificateless blind proxy signature algorithm based on PSS standard model," in *Proc. 5th Int. Conf. Cyber Security Intell. Analytics*, 2023, pp. 290–299.
[31] J. He and X. Mou, "Security analysis and improvement of certificateless proxy blind signature scheme," *Data Commun.*, vol. 5, pp. 20–23, 2020.
[32] A. Steane, "Quantum computing," *Rep. Prog. Phys.*, vol. 61, no. 2, p. 117, 1998.
[33] A. Khang, K. C. Rath, N. Panda, and A. Kumar, "Quantum mechanics primer: Fundamentals and quantum computing," in *Applications and Principles of Quantum Computing*. Hershey, PA, USA: IGI Glob. Sci. Publ., 2024, pp. 1–24.
[34] A. V. Aho and J. D. Ullman, *Foundations of Computer Science*. Comput. Sci. Press, 1992.
[35] L. Zhang and Y. Sang, "Proxy blind signature scheme from lattice basis delegation," *Int. J. Adv. Comput. Technol.*, vol. 4, no. 21, pp. 329–336, 2012.
[36] C. Li, G. Xu, Y. Chen, and J. Li, "A new anti-quantum proxy blind signature for blockchain-enabled Internet of Things," *Comput., Mater. Continua*, vol. 61, no. 2, p. 711, 2019.
[37] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, 2008, pp. 197–206.
[38] F. Li, M. Yang, Z. Song, P. Wang, and G. Li, "Post-quantum secure identity-based proxy blind signature scheme on a lattice," *Entropy*, vol. 25, no. 8, p. 1157, 2023.
[39] Y. Yan, M. Jiang, Y. Kong, and H. Ge, "Efficient identity-based proxy blind signature scheme over lattices," in *Proc. 2nd Int. Conf. Comput., Vis. Intell. Technol. (ICCVIT)*, 2024, pp. 1–7.
[40] F. Wu, W. Yao, X. Zhang, W. Wang, and Z. Zheng, "Identity-based proxy signature over NTRU lattice," *Int. J. Commun. Syst.*, vol. 32, no. 3, 2019, Art. no. e3867.
[41] H. Zhu, Y.-A. Tan, L. Zhu, X. Wang, Q. Zhang, and Y. Li, "An identity-based anti-quantum privacy-preserving blind authentication in wireless sensor networks," *Sensors*, vol. 18, no. 5, p. 1663, 2018.
[42] S. Singh, S. Rawal, S. Padhye, and N. Tiwari, "Identity based proxy blind signature scheme using NTRU lattices," *Inf. Comput.*, vol. 304, May 2025, Art. no. 105284.
[43] Å. Björck, "Numerics of Gram-Schmidt orthogonalization," *Linear Algebra Appl.*, vol. 197, pp. 297–316, Jan./Feb. 1994.
[44] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proc. 29th Annu. Int. Conf. Theory Appl. Cryptogr. Techn. Adv. Cryptol. (EUROCRYPT)*, 2010, pp. 1–23.
[45] C. Herrmann, D. Pickering, and M. Roddy, "A geometric description of modular lattices," *Algebra Universalis*, vol. 31, no. 3, pp. 365–396, 1994.
[46] C. Chuengsatiansup, T. Prest, D. Stehlé, A. Wallet, and K. Xagawa, "ModFalcon: Compact signatures based on module-NTRU lattices," in *Proc. 15th ACM Asia Conf. Comput. Commun. Security*, 2020, pp. 853–866.
[47] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2012, pp. 700–718.
[48] Z. Yang, D. H. Duong, W. Susilo, G. Yang, C. Li, and R. Chen, "Hierarchical identity-based signature in polynomial rings," *Comput. J.*, vol. 63, no. 10, pp. 1490–1499, 2020.
[49] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems," *J. ACM*, vol. 38, no. 3, pp. 690–728, 1991.
[50] R. del Pino and S. Katsumata, "A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling," in *Proc. 42nd Annu. Int. Cryptol. Conf. (CRYPTO)*, 2022, pp. 306–336.
[51] Y. Zhou, S. Dong, and Y. Yang, "A lattice-based identity-based proxy partially blind signature scheme in the standard model," *Netinfo Security*, vol. 21, no. 3, pp. 37–43, 2021.

Quanrun Li received the Ph.D. degree from the School of Cyber Science and Engineering, Wuhan University, Wuhan, China, in 2023.

He is currently a Lecturer with the School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou, China. His research interests include lattice-based cryptography and network security.

Jian Shen received the Ph.D. degree from Chosun University, Gwangju, South Korea, in 2012.

Since 2012 to 2022, he was a Professor with the School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, China. He is currently working as a Professor with the School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China. His research interests include public cryptography, cloud computing and security, data auditing and sharing, and information security systems.

Chao Lin received the Ph.D. degree from the School of Cyber Science and Engineering, Wuhan University, Wuhan, China, in 2020.

He is currently with the College of Cyber Security, Jinan University, Guangzhou, China. His main research interests include applied cryptography and blockchain technology.

Zhichao Wang is currently pursuing the master's degree with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China.

His research interests include applied cryptography and network security.

Debiao He (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2009.

He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. His main research interests include cryptography and information security, in particular, cryptographic protocols.