entropy

MDPI

Article

Identity-Based Proxy Signature with Message Recovery over NTRU Lattice

Faguo Wu [1,2,3,4,*], Bo Zhou [3] and Xiao Zhang [1,3,5]

[1] Key Laboratory of Mathematics, Informatics and Behavioral Semantics (LMIB), Beihang University, Beijing 100191, China
[2] Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
[3] Zhongguancun Laboratory, Beijing 100194, China
[4] Bejing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing 100191, China
[5] School of Mathematical Sciences, Beihang University, Beijing 100191, China
* Correspondence: faguo@buaa.edu.cn

Abstract: Proxy signature is one of the important primitives of public-key cryptography and plays an essential role in delivering security services in modern communications. However, existing post quantum proxy signature schemes with larger signature sizes might not be fully practical for some resource-constrained devices (e.g., Internet of Things devices). A signature scheme with message recovery has the characteristic that part or all of the message is embedded in the signature, which can reduce the size of the signature. In this paper, we present a new identity-based proxy signature scheme over an NTRU lattice with message recovery (IB-PSSMR), which is more efficient than the other existing identity-based proxy signature schemes in terms of the size of the signature and the cost of energy. We prove that our scheme is secure under a Short Integer Solution (SIS) assumption that is as hard as approximating several worst-case lattice problems in the random oracle model. We also discussed some application scenarios of IB-PSSMR in blockchain and Internet of Things (IOT). This paper provides a new idea for the design of lattice signature schemes in low resource constrained environments.

Keywords: lattice-based cryptography; proxy signature; message recovery; post quantum resistant

Citation: Wu, F.; Zhou, B.; Xiao, Z. Identity-Based Proxy Signature with Message Recovery over NTRU Lattice. *Entropy* **2023**, *25*, 454. https://doi.org/10.3390/e25030454

Academic Editor: Ivan B. Djordjevic

Received: 6 February 2023
Revised: 28 February 2023
Accepted: 3 March 2023
Published: 4 March 2023

1. Introduction

Proxy signature scheme is an emergency backup strategy of digital signatures, which can designate an agent to continue to perform signature verification in the absence of the signer. It was first proposed by Mambo, Usuda, and Okamoto et al. [1] in 1996. Subsequently, proxy signatures have been widely used in many scenarios, such as anonymous voting, electronic cash, mobile agents, etc. In the design of the construction scheme, most of the construction ideas are based on the difficult problems of traditional number theory, such as the difficult problems of (Elliptic Curve) discrete logarithms and factorization of large integers [2,3]. However, in the era of quantum computers, we need to find solutions based on other difficult problems, because these traditional schemes will be cracked by quantum algorithms in polynomial time [4]. Under this threat, many scholars began to study post quantum cryptography to prevent many important cryptosystems from failing directly after the advent of quantum computers. In the specific structure, there are mainly the following categories: lattice cryptography, multivariable cryptography, code-based cryptography, and Hash-based cryptography. Accordingly, some proxy signatures with post quantum security have been proposed, such as [5–9].

Lattice-based signature schemes have attracted many scholars' attention, as their difficulty assumptions rely on some math problems that have been widely studied and come with uniquely strong security guarantees where lattice cryptosystems, on average

(i.e., with randomly chosen keys), are as hard as the hardest problem of the underlying lattice problem [10]. Furthermore, In lattice cryptography, the operations involved in key generation, encryption, or signature usually involve only vector multiplication or modular addition over the integer ring, which makes the implementation of the scheme relatively simple. However, most lattice-based proxy signatures have large signature sizes, which makes lattice-based proxy signatures unsuitable in resource-constrained environments. Reducing the signature length is the most difficult problem in the practical application of lattice signatures, and how to solve and improve this problem is a critical question.

Traditional digital signature schemes usually need to bind messages and signatures to facilitate verifiers to verify them. This may incur additional bandwidth costs, especially when the message and signature sizes are relatively large. Scholars began to think about how to compress the size of messages and signatures as much as possible to reduce bandwidth consumption. The concept of message recovery was born in this case. Through message recovery, messages will be embedded in the signature. The sender sends the embedded signature to the receiver. After receiving the signature, the receiver can recover the original message from the signature and then perform signature verification. This construction method is very suitable for environments where signature size is required or bandwidth is limited [11,12]. In 1993, Nyberg and Ruppel modified the Digital Signature Algorithm (DSA) to support message recovery. It was the first signature scheme to support message recovery [13]. This has caused many scholars to pay attention to message recovery. Based on the lattice-based signature scheme of Lyubashevsky et al. [14], Tian et al. [15] constructed a scheme supporting message recovery on the lattice, allowing them to have more advantages in communication bandwidth than Lyubashevsky et al., but Tian et al.'s scheme does not support proxy for signing rights. In 2017, Faguo Wu et al. [16] considered the problem of signature authority proxy and constructed the first lattice based proxy signature scheme using public key infrastructure. In addition, their scheme supports message recovery, and then has a good performance in communication overhead. In 2019, Xiuhua Lu et al. [17] considered identity-based settings and constructed a proxy signature with message recovery over lattices. However, Refs. [16,17] are based on inefficient lattice structures, and these schemes are trapped in large signature sizes. People naturally think about how to construct efficient schemes with lattices. As far as we know, the NTRU lattice is the most efficient lattice. At present, it is still an open question whether the NTRU lattice can be used to construct a signature scheme with message recovery.

In terms of signature schemes designed based on quantum computing, Feng et al. [18] proposed a new quantum group signature scheme to enhance the non-repudiation of signatures. Lu et al. [19] proposed a verifiable arbitration quantum signature scheme based on controlled quantum teleportation, which can realize eavesdropping detection and identity authentication. Chen et al. [20] proposed a quantum multi-proxy blind signature based on cluster states to achieve blindness, non-repudiation and unforgeability. Feng et al. [21] studied an arbitrated quantum signature protocol based on boson sampling, which can resist forgery attack and denial attack. Feng et al. [22] proposed a quantum signature scheme for teleportation arbitration based on quantum walks, in which the entangled state is generated at the signature stage through quantum walks.

For the concrete application, Fang et al. [23] surveyed the application of proxy signatures in blockchain and investigated their usage in payment and integrity verification. In order to meet the challenges of data authentication and integrity in the Internet of Things environment, Verma et al. [24] proposed the first certificate-based proxy signature scheme without pairing. The proposed scheme is suitable for the Internet of Things in terms of computational cost. In the edge computing environment of the Internet of Things, resources are usually limited. Zhang et al. [25] proposed an ID-PRS scheme in the architecture of the Internet of Things, which also does not use pairing operations with high resource consumption, and supports non-interactive design. To address security and privacy issues in the Unmanned Aerial Vehicles (UAV) environment and mitigate various attacks, Verma

and Singh et al. [26] proposed a short proxy signature scheme based on certificate setting, which has advantages in signature length and computational efficiency.

In this paper, inspired by the lattice-based signature schemes [15,16,27,28], we first propose an identity-based proxy signature with message recovery over the NTRU lattice. In the random oracle model, our scheme can achieve delegation information and signature existential unforgeability under adaptive chosen warrant and identity attacks. Since our signature scheme adopts message recovery technology, compared with some existing proxy signature schemes, our scheme has better performance in communication overhead and signature size. Finally, when we consider the actual application [29], we find that this scheme performs well in terms of energy consumption, which means that our scheme is very suitable for resource constrained and low bandwidth environments. Due to the hardness assumption of SIS over the NTRU lattice, we formally constructed a lattice-based message recovery proxy signature scheme that can provide post quantum security in the quantum era.

The rest of the article is arranged as follows. In Section 2, we provide necessary preliminaries of our scheme. In Section 3, we give a detailed description of the syntax model and security model of our identity-based proxy signature with message recovery. In Section 4, we formally show how we construct the basic message recovery proxy signature. In Section 5, we present the formal security analysis of our scheme. In Section 6, we introduce detailed comparisons between our scheme and some existing proxy schemes. In Section 7, we discuss some application scenarios of our proposed IB-PSSMR scheme. Finally, we conclude our paper in Section 8.

2. Preliminary Knowledge

2.1. Notations

In this article, we agree that these tokens represent the following specific meanings:

- $\|v\|_p$ denotes the l_p norm of v.
- $M^{n\times(k_1+k_2)} = M_1^{n\times k_1} \parallel M_2^{n\times k_1}$ denotes the concatenation of Matrices M_1, M_2.
- $\mid x \mid$ indicates the length of x under binary representation.
- $\mid x \mid^{l_1}$ denotes the first left l_1 bits of x.
- $\mid x \mid_{l_2}$ denotes the first right l_2 bits of x.
- $x \parallel y$ denotes string concatenation. It means append string y at the behind of string x

2.2. NTRU Lattice

Let $\mathcal{R}_q$ be the ring $\mathbb{Z}_{\|}[x]/(x^N+1)$, and f,g be the polynomials in $\mathcal{R}_q$. Let h be the polynomial convolution of f^{-1} and g. In other words,

$$h = f^{-1}g \bmod (X^N+1) \tag{1}$$

where $f = \sum_{i=0}^{N-1} f_i x^i$ and $g = \sum_{i=0}^{N-1} g_i x^i$. The NTRU lattice associated with h and q is

$$\bigwedge_{h,q} = \{(u,v) : u + v * h \mod q = 0\} \tag{2}$$

$\bigwedge_{h,q}$ is a full rank lattice in $\mathbb{Z}^{2N}$ generated by the rows of

$$\mathbf{A_{h,q}} = \begin{pmatrix} \mathcal{A}_{\mathcal{N}}(h) & \mathbf{I_N} \\ q\mathbf{I_N} & \mathbf{O_N} \end{pmatrix} \tag{3}$$

where $\mathcal{A}_{\mathcal{N}}(h)$ is an anticirculant matrix whose ith row consists of the coefficients of the polynomial $hx^i \bmod (X^N+1)$. Additionally I_N is the $N \times N$ unit matrix, $\mathbf{O_N}$ is the $N \times N$ null matrix. We emphasize that NTRU lattices have some excellent properties: their Gram–Schmidt norm can be small and they can be computed quickly.

Definition 1. *Given integers q, m, n and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the $'q-ary'$ lattices are defined as follows*

$$\Lambda_q(\mathbf{A}) = \{x \in \mathbb{Z}^m : x = \mathbf{A}^T s \mod q, \text{ for some } s \in \mathbb{Z}_q^n\}$$
$$\Lambda_q^{\perp}(\mathbf{A}) = \{x \in \mathbb{Z}^m : \mathbf{A}^T x = 0 \mod q\}$$

$\Lambda_q(\mathbf{A})$ and $\Lambda_q^{\perp}(\mathbf{A})$ are dual to each other.

2.3. Gaussian on Lattice

In this section, we introduce an algorithm to sample the discrete Gaussian distribution, and the output result is a vector obeying the discrete Gaussian distribution. As shown in Algorithm 1.

Algorithm 1 GaussianSampler

Input: Lattice Λ basis B, standard deviation σ, center $c \in \mathbb{Z}^N$
Output: Vector v sampled in $D_{\Lambda,\sigma,c}$

$v_n \leftarrow 0$
$c_n \leftarrow c$
for $i = n, n-1, \cdots, 1$ **do**
  $c_i^{'} \leftarrow \langle c_i, \widetilde{b_i} \rangle / \| \widetilde{b_i} \|^2$
  $\sigma_i^{'} \leftarrow \| \widetilde{b_i} \|$
  $z_i \leftarrow SampleZ(c_i^{'}, \sigma_i^{'})$
  $c_{i-1} \leftarrow c_i - z_i b_i$
  $v_{i-1} \leftarrow v_i - z_i b_i$
end for
return v_0

The subalgorithm *SampleZ* samples a 1-dimensional Gaussian $D_{\mathbb{Z}^N,\sigma,c}$. There are various techniques for 1-dimensional discrete Gaussian sampling, such as the inverse method [30], the Knuth–Yao algorithm [31], rejection sampling [32] and discrete ziggurat algorithms [33].

According to Lyubashevsky's discussion on Lattice trapdoor [28] construction, consider the discrete Gaussian distribution in dimension m and let its standard deviation be σ, he proposed some important properties of Discrete Gaussian distribution. We refer it as Lemma 1.

Lemma 1. *$\forall \sigma > 0$ and $m \in \mathbb{Z}$*
(1) $Pr[x \in D_\sigma^1 : \mid x \mid > 12\sigma] < 2^{-100}$;
(2) $Pr[x \in D_\sigma^m : \| x \| > 2\sigma\sqrt{m}] < 2^{-m}$;
(3) For any $v \in \mathbb{Z}^m$ and any positive real α, if $\sigma = \omega(\| v \| \sqrt{\log m})$, then we have the following probability relation.

$$Pr[x \in D_\sigma^m : D_\sigma^m(\mathbf{x}) / D_{\sigma,v}^m = o(1)] = 1 - 2^{\omega \log m} \tag{4}$$

Additionally $\omega(.)$ is the non-asymptotic tight lower bound. More specifically, for a given quantity relationship, If $\sigma = \alpha \| v \|$, we can obtain the following inequality relation.

$$Pr[x \in D_\sigma^m : D_\sigma^m(\mathbf{x}) / D_{\sigma,v}^m < e^{12/\alpha + 1/(2\alpha^2)}] > 1 - 2^{-100} \tag{5}$$

2.4. Rejection Sampling Technique

The Rejection Sampling Technique [10] is mainly used to eliminate the relationship between the signing key and output signature. The algorithm is described below.

If the signer follows the steps in Algorithm 2, then the distribution of the outputted signatures is $min(\frac{D_\sigma^m(\mathbf{z})}{MD_{\mathbf{Sc},\sigma}(\mathbf{z})}, 1)$ and the expected number of times that this process will output a signature is M.

Algorithm 2 Rejection sampling technique

Input: Message u, a matrix A randomly sampled from $\mathbb{Z}_q^{m\times n}$, **S**(signature key) sampled from $\{-d, \cdots, 0, \cdots, d\}^{m\times k}$, $H : \{0,1\}^* \to \{v : v \in \{-1,0,1\}^k, \| v \| < \kappa\}$, where $d \ll q^{n/m}$, $k \in \mathbb{Z}$ and $\ll m$, κ is constant and $2^\kappa \cdot \binom{k}{\kappa} \geq 2^{100}$. Then there exists a constant $M = O(1)$.

Output: Vector **z** and **c**

1: Obtain **y** randomly from D_σ^m
2: $\mathbf{c} = H(A\mathbf{y}, u)$
3: $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$ **return** $(\mathbf{z}, \mathbf{c})$ with probability $min(\frac{D_\sigma^m(\mathbf{z})}{MD_{\mathbf{Sc},\sigma}(\mathbf{z})}, 1)$

2.5. Hardness Assumption

We assume the SIS problem is hard in the NTRU lattice, and referring to [34], when we choose f and g in key generation properly, the distribution of $h = f^{-1}g$ and uniform distribution of R^* are statistically close to each other, which means they are indistinguishable. Here we recall the definition of the **SIS** problem.

Definition 2. *(Small Integer Solution problem (**SIS**)) Let n and q be integers, where n stands for the security parameter. Typically q is a polynomial of n. Let $\beta > 0$. Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ where m also satisfies $m = poly(n)$, the goal is to find a non-zero vector $\mathbf{e} \in \mathbb{Z}^m$, such that $\mathbf{Ae} = 0 \mod q$ and $\| \mathbf{e} \| < \beta$.*

Definition 3. *Given f, g, h in NTRU's key pair generation, n, q, β is defined the same as in Definition 2. The **SIS** problem over NTRU lattice is to find a non-zero vector (z_1, z_2), such that it satisfies $\mathbf{A}_{h,q}(z_1, z_2) = 0 \ mod\ q$ and $\| (z_1, z_2) \| < \beta$.*

Assume that (s_1, s_2) is any of the vectors in the $A_{h,q}$, the $\gamma -$ **SVP** problem on the $A_{h,q}$ is to find the vector (z_1, z_2) satisfy $\| (z_1, z_2) \| \leq \gamma \| (s_1, s_2) \|$, that is, $\| (z_1, z_2) \| \leq \gamma\theta$. Among which θ is the shortest length of the vector in lattice $A_{h,q}$. Therefore, when $\gamma = \beta/\theta$, solving **SIS** over the NTRU lattice is as hard as solving the shortest vector problem in the NTRU lattice. Hence, we claim that our proposed scheme also relies on the hardness of $\gamma -$ **SVP**. Note that the $\gamma -$ **SVP** problem is **NP**-hard when the approximate factor $\gamma < 1 + 1/n^\varepsilon$ [35].

2.6. Message Recovery

Message recovery is a function extension of the signature scheme, allowing all or part of the messages to be embedded in the signature. The key generation, signature, verification algorithms, and message recovery process are shown in the Figure 1.

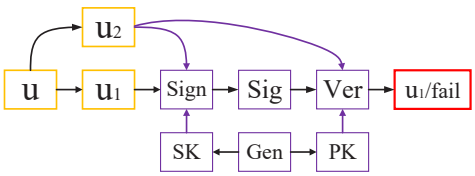

Figure 1. Signature with message recovery.

Gen, Sign, and Ver are the Key generation algorithm, signature and verification algorithm, SK is the secret key and PK is the public key. Message u to be signed is divided

into two parts $u = u_1 \parallel u_2$. u_1 is the recoverable part that is embedded in the signature and can be recovered from the signature during the verification process, and the non-recoverable part u_2 can be sent or stored with the signature.

3. Syntax and Security Model for Identity-Based Proxy Signature Scheme with Message Recovery

In this section, we will first give the syntax model, i.e, we describe the participants in our scheme, and the algorithms in our scheme. Then, we introduce the security model of our lattice-based proxy signature scheme with message recovery(IB-PSSMR).

3.1. Syntax

Definition 4. *There are four types of participants in our identity-based proxy signature with message recovery over the NTRU lattice:*

- *Original signer with* $\boldsymbol{ID}_o$*;*
- *Proxy signer* $\boldsymbol{ID}_p$*;*
- *Verifier;*
- *Key generation center (KGC) in the system.*

*Our scheme consists of six probabilistic polynomial-time (PPT) algorithms (**Setup**, **KeyExtract**, **DelGen**, **DelVer**, **Psign**, and **Pver**), and their roles are as follows:*

1. ***Setup**: The algorithm **Setup** takes a security parameters N as input, and then it outputs the system's public parameters par, KGC's public and secret key (mpk, msk), that is $(par, (msk, mpk)) \leftarrow \boldsymbol{Setup}(n)$.*
2. ***KeyExtract**: The algorithm **KeyExtract** takes the system's public parameters par, KGC's secret key msk and public key mpk, user's identity (i.e., user's public key pk) $\boldsymbol{ID}_u$ as input, and then it outputs the user $\boldsymbol{ID}_u$'s secret key sk_{ID}, that is, $sk_{ID} \leftarrow \boldsymbol{KeyExtract}(par, msk, \boldsymbol{ID}_u)$.*
3. ***DelGen**: The algorithm **DelGen**'s input consists of the system's public parameters par, KGC's public key mpk, a warrant W where $W = (pk_{ID_o}, pk_{ID_p}, T)$, T is valid time period of W, original signer's secret and public key (sk_{ID_o}, pk_{ID_o}), original signer computes the delegation, it outputs the delegation information d_g, that is, $\{d_g\} \leftarrow \boldsymbol{DelGen}(par, W, mpk, sk_{ID_o}, pk_{ID_o})$.*
4. ***DelVer**: On input the system's public parameters par, KGC's public key mpk, original signer's public key pk_{ID_o}, warrant W and its delegation d_g, he verifies the legality of delegation information d_g, If delegation d_g satisfied, the output is 1, and the delegation is accepted; otherwise, the output is 0, and the delegation is rejected, that is, $\{0,1\} \leftarrow \boldsymbol{DelVer}(par, W, d_g, mpk, pk_{ID_o}, pk_{ID_p})$.*
5. ***Psign**: Given the system's public parameters par, KGC's public key mpk, original signer's public key pk_{ID_o}, proxy signer's secret and public key (sk_{ID_p}, pk_{ID_p}), delegation key (sk_d, pk_d), warrant W and delegation information d_g, and the message m to be signed, the algorithm **Psign** outputs the identity-based proxy signature(IB-PS) on behalf of the original signer, that is, $sig \leftarrow \boldsymbol{Psign}(par, m, W, mpk, pk_{ID_o}, sk_{ID_p}, pk_{ID_p}, sk_d, pk_d)$.*
6. ***Pver**: For a verifier in our IB-PSSMR system, he first recovers the message m embedded in the signature sig. Then, the algorithm **Pver** takes the public key pk_{ID_o} of the original signer, the public key pk_{ID_p} of the proxy signer, and the public delegation key pk_d as input. if the proxy signature is valid, output 1, or output 0 if it is invalid, that is $\{m, \{0,1\}\} \leftarrow \boldsymbol{Pver}(par, sig, pk_{ID_o}, pk_{ID_p})$.*

Definition 5. *Given security parameters n, to make our scheme IB-PSSMR work correctly, the six PPT algorithms should meet the following rules*

$(par, (msk, mpk)) \leftarrow \boldsymbol{Setup}(n)$
$sk \leftarrow \boldsymbol{KeyExtract}(par, msk, ID)$
$\{sk_d, pk_d, d_g\} \leftarrow \boldsymbol{DelGen}(par, W, mpk, sk_{ID_o}, pk_{ID_o})$
$\{0,1\} \leftarrow \boldsymbol{DelVer}(par, W, d_g, mpk, sk_d, pk_d, pk_{ID_o}, pk_{ID_p})$
$sig \leftarrow \boldsymbol{Psign}(par, m, W, mpk, pk_{ID_o}, sk_{ID_p}, pk_{ID_p})$
$\{m, \{0,1\}\} \leftarrow \boldsymbol{Pver}(par, sig, pk_{ID_o}, pk_{ID_p})$
the above-mentioned algorithms hold with overwhelming probability.

3.2. Security Model for IB-PSSMR

For the security issue of identity-based proxy signature scheme with message recovery (IB-PSSMR) over NTRU lattice, there are two things we should concern about. First, the delegation is the proxy signer's signature on the message m, which is made on behalf of the original signer. Second, the warrant is a kind of timestamp restriction of message and contains the valid period of time. Considering this, Unforgeability, Verifiability, Strong identifiability, Strong undeniability, and Key dependence are naturally satisfied. Therefore, the security model of this IB-PSSMR over NTRU lattice is existential unforgeable under adaptive chosen-message attacks. We define the security model of our IB-PSSMR by a game, or an experiment, run between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$(forger).

In regard to the unforgeability of our IB-PSSMR over NTRU lattice, we should take two types of adversary into consideration:

Type(i): Adversary $\mathcal{A}$ can obtain access to the original signer's public key pk_{ID_o}, proxy signer's public key pk_{ID_p},original signer's secret key sk_{ID_o}.

Type(ii) : Adversary $\mathcal{A}$ can not obtain access to the original signer's secret key sk_{ID_o}, proxy signer's secret key sk_{ID_p}.

It is evident that the adversary in **Type**(i) is more powerful than the adversary in **Type**(ii), thus we will only consider the **Type**(i) adversary.

The security game of the IB-PSSMR is defined by the interactions between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. Additionally, the interactions consist of the following phases:

1. Initial Phase: the challenger $\mathcal{C}$ runs the **Setup**(n) algorithm to generate the system public parameters par and then $\mathcal{C}$ sends them to the adversary $\mathcal{A}$.
2. Query Phase: in the Query Phase, the adversary $\mathcal{A}$ can adaptively issue some query (also known as query the oracles). The number of queries is polynomial bounded.
   - **KeyExtract**-query: given an ID, the adversary $\mathcal{A}$ can issue a query to obtain the corresponding secret key. The challenger $\mathcal{C}$ runs the algorithm $sk_{ID} \leftarrow$ **DelGen**$(par, W, mpk, sk_{ID_o}, pk_{ID_o})$, and returns $\mathcal{A}$ with sk_{ID}.
   - **DelGen**-query: for some interested delegation information d_g, the adversary $\mathcal{A}$ issues query with two secret key corresponding to the identity ID_o and ID_p as input. Once upon receiving the query, the challenger $\mathcal{C}$ runs $d_g \leftarrow$ **DelGen**$(par, W, mpk, sk_{ID_o}, pk_{ID_o})$. Additionally, $\mathcal{C}$ returns d_g to $\mathcal{A}$.
   - **Psign**-query: if $\mathcal{A}$ is interested in the proxy signature of message m under ID_p, he issues such a query to the challenger. $\mathcal{C}$ runs the algorithm sig $\leftarrow$ **Psign**$(par, m, W, mpk, pk_{ID_o}, sk_{ID_p}, pk_{ID_p})$, and delivers sig to $\mathcal{A}$.
3. Forgery Phase: through the query phase above, the adversary $\mathcal{A}$ tries to forge a proxy signature to win the game. Given a message m and an identity ID_p as the proxy signer, $\mathcal{A}$ needs to generate a valid sig to make it pass the verification. The following conditions should naturally be satisfied:
   (a) **Pver**$(par, pk_{ID_o}, pk_{ID_p}) = 1$.
   (b) In the **Psign**-query phase, m has never been signed.
   (c) In the **KeyExtract**-query phase, the secret key of ID_p has not been queried.

Definition 6. *If the advantage of any PPT adversary $\mathcal{A}$ wins the security game above is negligible, then the Identity-based proxy signature with message recovery(IB-PSSMR) over NTRU lattice is regarded as existential unforgeable.*

4. Our Identity-Based Proxy Signature Scheme with Message Recovery

The identity-based proxy signature scheme with message recovery (IB-PSSMR) over NTRU lattice we proposed is discussed in this section. There are four participants in our scheme:

- A trusted third party KGC,
- An original signer with ID_o,
- A proxy signer with ID_p,

- A verifier.

Additionally, our scheme IB-PSSMR over NTRU lattice consists of six probabilistic polynomial time algorithms (**Setup**, **KeyGen**, **DelGen**, **DelVer**, **Psign**, and **Pver**), where:

1. **Setup**: the **Setup** algorithm run by KGC. It takes a system security parameter λ as the algorithms' input. Assume $q \geq 3$, λ, N be positive integers. The **Setup** algorithm will do the following steps:
   - Choose hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^N$, $H_2 : \mathbb{Z}_q^n \to \{0,1\}^{l_1+l_2}$, $H_3 : \{0,1\}^* \to \{0,1\}^{N\times N}$, $l_1, l_2 \in \mathbb{N}$, H_2, H_3 are seen as a random oracle.
   - Select two encoding functions $F_1 : \{0,1\}^{l_2} \to \{0,1\}^{l_1}$, $F_2 : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$.
   - KGC starts the algorithm **MasterKeygen** to output the system's master key (msk, mpk), which is described in Algorithm 3.
   - Finally, KGC publishes $par = (N, q, H_1, H_2, H_3)$ as public parameters of our IB-PSSMR system.
2. **KeyExtract**: KGC takes the public parameters par and system's master secret key msk as the algorithm's input, then KGC works as follows:
   - The system's participants original signer and proxy signer request their secret key from KGC, and offer their identity ID_o and ID_p, respectively.
   - KGC first checks whether these identities exist in the identity list **IDLIST**. If so, **KeyExtract** request can be terminated, otherwise, KGC runs **GaussianSampler** $(c, \sigma, (H_1(ID_o), 0))$ to obtain ID_o's secret key $\mathbf{sk_o} = (\mathbf{s_1}, \mathbf{s_2})$ and runs **Gaussian Sampler**$(c, \sigma, (H_1(ID_p), 0))$ to obtain ID_p's secret key $\mathbf{sk_p} = (\mathbf{s_3}, \mathbf{s_4})$, where $\mathbf{s_1} + \mathbf{s_2}h = H_1(ID_o)$ and $\mathbf{s_3} + \mathbf{s_4}h = H_1(ID_p)$.
   - KGC sends sk_p to the proxy signer and sk_o to the original signer by a a secure authenticated channel.
3. **DelGen**: original signer generates the delegation on warrant W where $W = (pk_{ID_o}, pk_{ID_p}, T)$, T is the valid time period of W, and delegation information d_g on W is described as Algorithm 4.
4. **DelVer**: when the proxy signer receives the warrant W and its delegation $d_g = (z_1, z_2)$, he first checks if $\| (\mathbf{z_1}, \mathbf{z_2}) \| \leq 2\sigma\sqrt{2N}$ and $H_2(hy_2 + y_1 - H_1(ID_O) * W, W)$ both are true. If the conditions hold, then proxy signer ID_p can take the warrant as his lawful authority from the original signer; otherwise, he should reject it.
5. **Psign**: after confirming the legitimacy of the signer, given a message u, the proxy signer with ID_p can generate a proxy signature for it by Algorithm 4.
6. **Pver**: given the public parameters par, for a a user in the system who wants to verify the legitimacy of the proxy signature, he performs the steps described in Algorithm 5.

Theorem 1. *The IB-PSSMR we proposed satisfies correctness.*

Proof. From the Algorithms 3–5's detailed construction, we can easily have the following equations.

$$\begin{aligned} &H_2(hz_{i+1} + z_i - H_1(ID)H_3(r, u_2)) \\ &= H_2(h(s_{i+1}C + y_2) + (s_iC + y_1) - (s_{i+1}h + s_i)C) \\ &= H_2(y_1 + y_2h) \\ &= \alpha \end{aligned}$$

the distribution of (z_{i+1}, z_i) and the distribution $D_{\mathbb{Z}^N,s}$ are statistically close to each other. By the Lemma 1, $\| z_i \| \leq 2\sigma\sqrt{N}$ with probability at least $1 - 2^{-m}$, that is, $\| (z_{i+1}, z_i) \| \leq 2\sigma\sqrt{2N}$ satisfied with overwhelming probability. Furthermore, $u_1' = F_1(u_1) \| (F_2(F_1(u_1)) \bigoplus u_1)$, we can recover $u_1 = |u_1'|_{l_2} \bigoplus F_2(|u_1'|^{l_1})$ with $F_1(u_1) = |u_1'|^{l_1}$ hold. □

Algorithm 3 Master Keygen

Input: Security parameter N, prime q, σ
Output: KGC's public key mpk and secret key msk.

Start Sample $f, g \in D_{Z^N,\sigma}$.
if $\| f \| > \sigma\sqrt{N}$ or $\| g \| > \sigma\sqrt{N}$ or $f \bmod q \notin R_q^*$ or $g \bmod q \notin R_q^*$ **then**
    Restart
end if
if $\max(\| (g, -f) \|, \| (\frac{g\overline{f}}{f\overline{f}+g\overline{g}}, \frac{g\overline{g}}{f\overline{f}+g\overline{g}}) \|) > 1.17\sqrt{g}$ **then**
    Restart
end if
$R_f = \text{resultant}(f, X^N + 1)$ and $R_g = \text{resultant}(g, X^N + 1)$, respectively. The resultant of f can be straightforwardly calculated as $\prod_{i=1}^{N-1} f(X^i) \pmod{\Phi(N)}$ where $\Phi(N)$ is the cyclotomic polynomial $\Phi(N) = 1 + X + X^2 + \cdots + X^{N-1}$. The details of the *resultant* operation can refer to [36]
Compute ρ_f, ρ_f satisfy $\rho_f f + k_f(X^N + 1) = R_f$, $\rho_g f + k_g(X^N + 1) = R_g$ by the Extended Euclidean Algorithm where k_f and k_g are integers.
if $(R_f, R_g) \neq 1$ **then**
    Restart
end if
Use the Extended Euclidean Algorithm to find α and β satisfy $\alpha R_f + \beta R_g = 1$, that is, we have $(\alpha\rho_f)f + (\beta\rho_g)g = 1 + k(x^N + 1)$.
Let $F = q\beta\rho_g$, $G = -q\alpha\rho_f$, then $f * G - g * F = q \pmod{X^N + 1}$
return The KGC's master public key $mpk = h = f^{-1}g$, KGC's master secret key $msk = B = \begin{pmatrix} \mathcal{A}_g & -\mathcal{A}_f \\ \mathcal{A}_G & -\mathcal{A}_F \end{pmatrix}$, where $\mathcal{A}_g$, $-\mathcal{A}_f$, $\mathcal{A}_G$ and $-\mathcal{A}_F$ are anti-circulant matrices, and their ith row consists of the coefficients of the polynomial $gx^i \mod (X^N + 1)$, $fx^i \mod (X^N + 1)$, $Gx^i \mod (X^N + 1)$ and $Fx^i \mod (X^N + 1)$, respectively.

Algorithm 4 Message recovery

Input: Private key $sk = (s_i, s_{i+1})$, message u
Output: Message recovery signature (z_i, z_{i+1})

Choose $y_1, y_2 \in D_{\mathbb{Z}^N,\sigma}$
Divide the message u into two parts $u = u_1 \| u_2$ and make $|u_1| = l_2$, if $|u| < l_2$ then let $u_2 = \perp$.
Compute $\alpha = H_2(y_1 + y_2 h)$.
Compute $u_1^{'} = F_1(u_1) \| (F_2(F_1(u_1)) \bigoplus u_1)$.
Compute $r = \alpha \bigoplus u_1^{'}$.
Compute $C = H_3(r, u_2)$
Compute $z_i = s_i C + y_1, z_{i+1} = s_{i+1} C + y_2$.
if Nothing is outputted **then**
    Restart
end if
return (u_2, z_i, z_{i+1}) on message m with probability $min(\frac{D_{\mathbb{Z}^N,\sigma}}{M D_{\mathbb{Z}^N,\sigma,sku}}, 1)$, where $M = O(1)$.

Algorithm 5 Pver

Input: r, z_i, z_{i+1}, u_2
Output: 0 or 1
Compute $\alpha = H_2(hz_{i+1} + z_i - H_1(ID)H_3(r, u_2))$
Compute $u_1' = r \bigoplus \alpha$
$u_1 = |u_1'|_{l_2} \bigoplus F_2(|u_1'|^{l_1})$
Compute $u = u_1 \parallel u_2$
if $\parallel (z_i, z_{i+1}) \parallel \leq 2\sigma\sqrt{2N}$, $F_1(u_1) = |u_1'|^{l_1}$ **then**
    Return 1
else
    Return 0
end if

5. Security Analysis

In this section, we give a formal proof to show that our proxy signature is unforgeable. If not, the adversary can break the hardness problem SIS in the NTRU lattice.

Theorem 2. *The proposed IB-PSSMR over NTRU lattice is existential unforgeable against adaptive chosen message and address attacks in the random oracle model under the hardness assumption of SIS problem over NTRU lattice.*

Proof. We prove the security of our scheme by contradiction. Suppose that if there is a PPT adversary $\mathcal{A}$ who can break our IB-PSSMR over NTRU lattice with non-negligible probability, we show that the adversary $\mathcal{A}$ can then solve the SIS problem over NTRU lattice.

The security game can be described between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. We simulate the interaction between challenger $\mathcal{C}$ and adversary $\mathcal{A}$ as follows:

Initial Taking λ as the security parameter, the algorithm $\mathcal{C}$ first randomly picks a matrix h, three secure hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^N$, $H_2 : Z_q^n \to \{0,1\}^{l_1+l_2}$, $H_3 : \{0,1\}^* \to \{0,1\}^{N\times N}$ and two encoding functions $F_1 : \{0,1\}^{l_2} \to \{0,1\}^{l_1}$, $F_2 : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$ then sends the public parameters *par*={h, H_1, H_2, H_3, F_1, F_2} to the adversary $\mathcal{A}$.

Queries: The adversary $\mathcal{A}$ issues the following queries adaptively.

- H_1-query: to make use of the H_1 oracle response, the challenger $\mathcal{C}$ builds a list L_0 to store the query response information. It is initialized as an empty set. Given the adversary's H_1 query with ID_i, $\mathcal{C}$ first check if it is in the list L_0. If there is a value corresponding to $H_1(\mathrm{ID}_i)$, then return it to the adversary. Otherwise the challenger randomly chooses $H_1(\mathrm{ID}_i) \in \mathbb{Z}_q^N$, then updates the H_1 list L_0 as $L_0 = (L_0, \{\mathrm{ID}_i, H_1(\mathrm{ID}_i)\})$, and finally outputs $H_1(\mathrm{ID}_i)$ as the response.
- H_2-query: the challenger $\mathcal{C}$ maintains the H_2 list which is a list of tuples $L_1 = (\alpha_i, y_{i1} + y_{i2}h)$, and the initial value is null, when the adversary $\mathcal{A}$ issues a H_2 query on a vector $y_{i1} + y_{i2}h \in \mathbb{Z}_q^N$, the challenger $\mathcal{C}$ looks it up in the H_2 list, if the challenger $\mathcal{C}$ finds a matched tuple $(\alpha_i, y_{i1} + y_{i2}h)$, he returns α_i to adversary $\mathcal{A}$ as the query response. If not, $\mathcal{C}$ randomly selects string $\alpha_i \in \{0,1\}^{l_1+l_2}$, then updates the H_2 list L_1 as $L_1 = (L_1, \{\alpha_i, y_{i1} + y_{i2}h\})$, and finally outputs α_i as the response.
- F_1-query: the challenger $\mathcal{C}$ maintains a F_1 list $L_2 = (u_{i1}, F_1(u_{i1}))$, and set it empty in the beginning. When there is a F_1 query for u_{i1} from the adversary $\mathcal{A}$, the challenger $\mathcal{C}$ first checks if it is in the L_2 list. If there is a corresponding pair $(u_{i1}, F_1(u_{i1}))$ in list L_2, then send $F_1(u_{i1})$ back to $\mathcal{A}$ as the query response. Otherwise, $\mathcal{C}$ randomly picks $F_1(u_{i1}) \in \{0,1\}^{l_1}$, then updates the list $L_2 = (L_2, (u_{i1}, F_1(u_{i1})))$, and finally outputs $F_1(u_{i1}) \in \{0,1\}^{l_1}$ as the response.
- F_2-query: the challenger $\mathcal{C}$ maintains a F_2 list $L_3 = (F_1(u_{i1}), F_2(F_1(u_{i1}))$, and set it empty in the beginning. When there is a F_1 query for u_{i1} from adversary $\mathcal{A}$, the challenger $\mathcal{C}$ firstly checks if it is in the L_4 list. If there is a corresponding pair $(F_1(u_{i1}), F_2(F_1(u_{i1}))$, return $F_2(F_1(u_{i1}))$, otherwise, challenger randomly chooses $F_2(F_1(u_{i1}) \in \{0,1\}^{l_2}$, then updates

the list $L_3 = (L_3, (F_1(u_{i1}), F_2(F_1(u_{i1}))))$, and finally outputs $F_2(F_1(u_{i1})) \in \{0,1\}^{l_2}$ as the response.

- H_3-query: the challenger $\mathcal{C}$ maintains a H_3 list $L_4 = (r_i, u_{i2}, \mathbf{C}_i)$, and also sets the list as an empty set in the initial phase. When there is a query for (r_i, u_{i2}), the challenger $\mathcal{C}$ firstly checks if it is in the list. If it exists, then return the corresponding array $(r_i, u_{i2}, \mathbf{C_i})$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ randomly selects vector $\mathbf{C}_i \in \{-1, 0, 1\}^{N \times N}$, then updates the list $L_4 = (L_4, (r_i, u_{i2}, \mathbf{C}_i))$, and finally outputs $\mathbf{C}_i$ as the response.
- *KeyExtract*-query: the challenger $\mathcal{C}$ maintains a *KeyExtract* list $L_5 = (ID_i, sk_{ID_i})$, and makes the list an empty set in the beginning. Now if the adversary $\mathcal{A}$ initiates a request for the private key associated with an identity ID_i, the challenger $\mathcal{C}$ checks if it is already in the L_5 list. If there exists the corresponding pair (ID_i, sk_{ID_i}), then the challenger $\mathcal{C}$ returns sk_{ID}. Otherwise $\mathcal{C}$ recovers the corresponding $(ID_i, H_1(ID_i))$ from the L_0 list, then $\mathcal{C}$ runs **GaussianSampler**$(c, \sigma, H_1(ID_i), 0))$ to obtain $sk_{ID_i} = (s_{i_1}, s_{i_2})$, then updates the list $L_5 = (L_5, ID_i, sk_{ID_i})$.
- *DelGen*-query: the challenger $\mathcal{C}$ maintains a *DelGen* list $L_6 = (y_{i_1}, y_{i_2}, u_{o_2}, z_{i_1}, z_{i_2})$ where warrant $W_i = u_{o_1} \parallel u_{o_2}$, When the adversary $\mathcal{A}$ issues a *DelGen* query for delegation of warrant W_i, the challenger $\mathbf{C_i}$ searches it in L_6 list first, if there exist corresponding tuple $(y_{i_1}, y_{i_2}, u_{o_2}, z_{i_1}, z_{i_2})$, return z_{i_1}, z_{i_2}, otherwise, the adversary $\mathcal{A}$ executes $z_{i_1} = s_{o_{i_1}}\mathbf{C_o} + y_{i_1}, z_{i+1} = s_{o_{i_2}}\mathbf{C_o} + y_{i_2}$ to obtain a valid delegation signature, then updates the list $L_6 = (L_6, y_{i_1}, y_{i_2}, u_{o_2}, z_{i_1}, z_{i_2})$.
- *Psign*-query: the challenger $\mathcal{C}$ maintains a *Psign* list $L_7 = (y_{i_3}, y_{i_4}, u_{p_2}, z_{i_3}, z_{i_4})$ where message $U = u_{p_1} \parallel u_{p_2}$, when the adversary $\mathcal{A}$ issues a *Psign* query for the proxy signature of message U, the challenger $\mathcal{C}$ searches it in the L_7 list first, if there exists a corresponding tuple $(y_{i_3}, y_{i_4}, u_{p_2}, z_{i_3}, z_{i_4})$, return (z_{i_3}, z_{i_4}). Otherwise, the adversary $\mathcal{A}$ executes $z_{i_3} = s_{p_{i_1}}\mathbf{C_p} + y_{i_3}, z_{i+1} = s_{p_{i_2}}\mathbf{C_p} + y_{i_4}$ to obtain a valid proxy signature, then updates the list $L_7 = (L_7, y_{i_3}, y_{i_4}, u_{p_2}, z_{i_3}, z_{i_4})$.

Forgery After the interactions and queries, the adversary $\mathcal{A}$ outputs a valid forgery $(u_{o_2}, u_{p_2}, z_{i_1}, z_{i_2}, z_{i_3}, z_{i_4})$ with non-negligible probability on warrant W, message U, original signer identity ID_o and proxy signer identity ID_p. We show that if $\mathcal{A}$ can do this forgery correctly then he is able to obtain a short non-zero solution of a SIS instance over NTRU lattice, i.e., the equation system $\mathbf{A}_{h,q}(z_1, z_2) = 0 \bmod q$ where $\parallel (z_1, z_2) \parallel < \beta$. The Queries phase can be executed again by $\mathcal{A}$. According to the Forking lemma in [37] to generate another valid signature $(u^*_{o_2}, u^*_{p_2}, z^*_{i_1}, z^*_{i_2}, z^*_{i_3}, z^*_{i_4})$.

$$H_2(h\mathbf{z_{i_2}} + \mathbf{z_{i_1}} - H_1(\mathrm{ID}_o)\mathbf{C_o}) = H_2(h\mathbf{z^*_{i_2}} + \mathbf{z^*_{i_1}} - H_1(\mathrm{ID}_o)\mathbf{C_o}^*) \tag{6}$$

$$H_2(h\mathbf{z_{i_4}} + \mathbf{z_{i_3}} - H_1(\mathrm{ID}_p)\mathbf{C_p}) = H_2(h\mathbf{z^*_{i_4}} + \mathbf{z^*_{i_3}} - H_1(\mathrm{ID}_p)\mathbf{C_p}^*) \tag{7}$$

The following equation is true unless we can find a collision of the hash function H_2, which is hard in the random oracl model. So we can ensure their preimage is same.

$$h\mathbf{z_{i_2}} + \mathbf{z_{i_1}} - H_1(\mathrm{ID}_o)\mathbf{C_o} = h\mathbf{z^*_{i_2}} + \mathbf{z^*_{i_1}} - H_1(\mathrm{ID}_o)\mathbf{C_o}^*$$

$$h\mathbf{z_{i_4}} + \mathbf{z_{i_3}} - H_1(\mathrm{ID}_p)\mathbf{C_p} = h\mathbf{z^*_{i_4}} + \mathbf{z^*_{i_3}} - H_1(\mathrm{ID}_p)\mathbf{C_p}^*$$

Rearranging the two sides in the two equations, we obtain

$$h(\mathbf{z_{i_2}} - \mathbf{z^*_{i_2}}) + \mathbf{z_{i_1}} - \mathbf{z^*_{i_1}} + H_1(\mathrm{ID}_o)(\mathbf{C^*_o} - \mathbf{C_o}) = 0$$

$$h(\mathbf{z_{i_4}} - \mathbf{z^*_{i_4}}) + \mathbf{z_{i_3}} - \mathbf{z^*_{i_3}} + H_1(\mathrm{ID}_p)(\mathbf{C^*_p} - \mathbf{C_p}) = 0$$

Since we have $\mathbf{s}_i + \mathbf{s}_{i+1}h = H_1(\mathrm{ID}_i)$. We obtain

$$h(\mathbf{z_{i_2}} - \mathbf{z^*_{i_2}}) + \mathbf{z_{i_1}} - \mathbf{z^*_{i_1}} + (\mathbf{s}_1 + \mathbf{s}_2 h)(\mathbf{C^*_o} - \mathbf{C_o}) = 0$$

$$h(\mathbf{z_{i_4}} - \mathbf{z^*_{i_4}}) + \mathbf{z_{i_3}} - \mathbf{z^*_{i_3}} + (\mathbf{s}_3 + \mathbf{s}_4 h)(\mathbf{C^*_p} - \mathbf{C_p}) = 0$$

Focusing on h, we have

$$h(\mathbf{z_{i_2}} - \mathbf{z^*_{i_2}} + \mathbf{s_{i_2}C^*_o} - \mathbf{s_{i_2}C_o}) + \mathbf{z_{i_1}} - \mathbf{z^*_{i_1}} + \mathbf{s_{i_1}C^*_o} - \mathbf{s_{i_1}C_o} = 0$$

$$h(\mathbf{z_{i_4}} - \mathbf{z^*_{i_4}} + \mathbf{s_{i_4}C^*_p} - \mathbf{s_{i_4}C_p}) + \mathbf{z_{i_3}} - \mathbf{z^*_{i_3}} + \mathbf{s_{i_3}C^*_p} - \mathbf{s_{i_3}C_p} = 0$$

Then, we write the equations in matrix form, which are

$$\begin{pmatrix} h \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{z_{i_2}} - \mathbf{z^*_{i_2}} + \mathbf{s_{i_2}C^*_o} - \mathbf{s_{i_2}} * \mathbf{C_o} & \mathbf{z_{i_1}} - \mathbf{z^*_{i_1}} + \mathbf{s_{i_1}C^*_o} - \mathbf{s_{i_1}} * \mathbf{C_o} \end{pmatrix} = \mathbf{0}$$

$$\begin{pmatrix} h \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{z_{i_4}} - \mathbf{z^*_{i_4}} + \mathbf{s_{i_4}C^*_p} - \mathbf{s_{i_4}} * \mathbf{C_p} & \mathbf{z_{i_3}} - \mathbf{z^*_{i_3}} + \mathbf{s_{i_3}C^*_p} - \mathbf{s_{i_3}} * \mathbf{C_p} \end{pmatrix} = \mathbf{0}$$

As $\| (\mathbf{z_i}, \mathbf{z^*_i}) \| \leq 2\sigma\sqrt{2N}$ and $\| (\mathbf{s_{i_1}}, \mathbf{s^*_{i_1}}) \| \leq s\sqrt{2N}$ with overwhelming probability. We obtain

$$\| (\mathbf{z_{i_2}} - \mathbf{z^*_{i_2}} + \mathbf{s_{i_2}C^*_o} - \mathbf{s_{i_2}} * \mathbf{C_o}, \mathbf{z_{i_1}} - \mathbf{z^*_{i_1}} + \mathbf{s_{i_1}C^*_o} - \mathbf{s_{i_1}} * \mathbf{C_o}) \| \leq (4\sigma + 4s\lambda)\sqrt{2N}$$

$$\| (\mathbf{z_{i_4}} - \mathbf{z^*_{i_4}} + \mathbf{s_{i_4}C^*_p} - \mathbf{s_{i_4}} * \mathbf{C_p}, \mathbf{z_{i_3}} - \mathbf{z^*_{i_3}} + \mathbf{s_{i_3}C^*_p} - \mathbf{s_{i_3}} * \mathbf{C_p}) \| \leq (4\sigma + 4s\lambda)\sqrt{2N}$$

Now if $(\mathbf{z_{i_2}} - \mathbf{z^*_{i_2}} + \mathbf{s_{i_2}C^*_o} - \mathbf{s_{i_2}} * \mathbf{C_o}, \mathbf{z_{i_1}} - \mathbf{z^*_{i_1}} + \mathbf{s_{i_1}C^*_o} - \mathbf{s_{i_1}} * \mathbf{C_o}) \neq 0$ and $(\mathbf{z_{i_4}} - \mathbf{z^*_{i_4}} + \mathbf{s_{i_4}C^*_p} - \mathbf{s_{i_4}} * \mathbf{C_p}, \mathbf{z_{i_3}} - \mathbf{z^*_{i_3}} + \mathbf{s_{i_3}C^*_p} - \mathbf{s_{i_3}} * \mathbf{C_p}) \neq 0$, it means that we can find an meaningful non-zero solution for a SIS instance in the NTRU lattice with overwhelming chance. Given Property 4 in [28] for Collision-Resistant preimage sampleable functions, the probability that algorithm $\mathcal{C}$ breaks the Short Integer Solution problem over the particular NTRU lattice is at least $(1 - 2^{\omega(\sqrt{\log N})})\varepsilon$.

Therefore, assuming we are in random oracle model (ROM), if there is a PPT adversary $\mathcal{A}$ that can break the proposed IB-PSSMR over NTRU lattice with a non-negligible probability ϵ. Then we can use the algorithm $\mathcal{A}$ to construct a new PPT algorithm $\mathcal{C}$ to find a solution for the SIS problem in NTRU lattice. Additionally, which can be reduced to SVP problem over the NTRU lattice. So, assume the hardness of SVP problem, we claim our IB-PSSMR scheme is unforgeable. Given there is no known quantum algorithm for SVP, we can that claim our IB-PSSMR is also quantum resistant.

Furthermore, it is not difficult to prove that our IB-PSSMR scheme is identifiability, strong undeniability, key dependence, and verifiability, for simplicity, we omit it here. □

6. Efficiency Analysis

At present, there are two kinds of security models for signature schemes, Random Oracle Model and Standard Model. Mostly, the more efficient lattice-based proxy signature schemes are those that proved secure in the random oracl model. Agrawal et al. [38] proposed a secure identity-based encryption scheme under the standard model, but their scheme is inefficient and can only encrypt one plaintext bit.

In this section, we will analyse some related proxy signature schemes and compare their metric with ours. We list the comparison of the signature length between our scheme and the related scheme under the same security parameter N setting, where $m > 5N\log q$, $\sigma = 12\lambda m\sqrt{m}\omega(\sqrt{\log N})$, W is the warrant, and U is the information to be signed.

From Table 1, the total length (signed message and signature) of scheme [39] is $| W |+| U |+4N\log(12\sigma)+2N(\log\lambda+1)=| u_{o2} | + | u_{p2} |+2l_2+4N\log(12\sigma)+2N(\log\lambda+1)$, the total length our message recovery signature scheme is $| u_{o2} | + | u_{p2} | +2 | r |+4N\log(12\sigma)=| u_{o2} | + | u_{p2} | +2l_1 + 2l_2+4N\log(12\sigma)$. Therefore, we make a proper

reduction of $2N(\log\lambda+1)-2l_1$ in the communication overhead compared with [39] which is based on the NTRU lattice without message recovery.

Table 1. Performance comparison among Refs. [39,40] and our scheme.

	Message Recovery	Delegation	Signature's Size
[39]	No	Yes	$\mid u_{o2} \mid + \mid u_{p2} \mid$+$2l_2$+4Nlog(12$\sigma$)+2N(log$\lambda$+1)
[40]	No	No	$N\lceil \log q \rceil$
Ours	Yes	Yes	$\mid u_{o2} \mid + \mid u_{p2} \mid + 2l_1 + 2l_2$+4Nlog(12$\sigma$)

Ducas et al. [40] proposed an efficient identity-based encryption (IBE) scheme based on NTRU lattice and a method to convert it into an identity-based signature (IBS) under the same framework. Compared with the scheme of [40], this paper adds the signature proxy authority and message recovery function. By constructing message recovery, in terms of transmission efficiency, our scheme can save communication bandwidth and only increase a small amount of computing resource consumption.

When we let security parameter $N = 512$, we present the concrete instances of communication overhead reduction between our scheme and [39] in Table 2.

Table 2. Approximate measure of some concrete parameter instance.

Parameter Size (N, Instance, q, k, λ, l_1)	Communication Overhead Reduction (Bits)
(512, 1, 227, 80, 28, 100)	2305
(512, 2, 225, 512, 14, 100)	1997
(512, 3, 233, 512, 14, 200)	1777

Furthermore, the energy consumption in transmission and computation is different. It is shown that a 32-bit computation requires less energy than a bit of transmission [29]. In our IB-PSSMR scheme, even if we make use of some more simpler computations, e.g., XOR and hash, in message recovery technology, we still obtain much less energy consumption than in the practical case [39].

Given the analysis above, we can conclude that the IB-PSSMR we refer to is more efficient than other lattice-based schemes in terms of communication and energy consumption.

7. Application of The IB-PSSMR

In this section, we discuss some application scenarios of our proposed IB-PSSMR scheme. Mostly, we will discuss its application in blockchain and Internet of Things.

For the proxy signature scheme, it is mainly about delegation authority. In the blockchain, the transfer of authority is often involved, such as transfer authority and certificate deposit authority [41]. In the cryptocurrency blockchain system, the private key of a wallet is usually held by a single node. However, in some cases, the currency of a wallet is publicly owned by an organization member, or it is necessary to give some proxy permissions to other nodes, which can exercise the same transfer permissions. At this time, the use of a proxy signature is needed. The frame diagram is shown in Figure 2. The wallet owning node will authorize the nodes within the organization with signature authority. The nodes that receive the legal proxy authorization can sign the transaction. After the signed transaction enters the transaction pool, it will be authenticated by the mining node to complete the confirmation of the transaction process. In the blockchain, to maintain the scalability of the blockchain, the block size of the blockchain will be strictly controlled. Therefore, the signature size of the transaction will also have an important impact on the performance of the blockchain. The IB-PSSMR scheme we proposed can compress the

size of the signature well and can be used as an alternative signature algorithm for the post-quantum blockchain design.

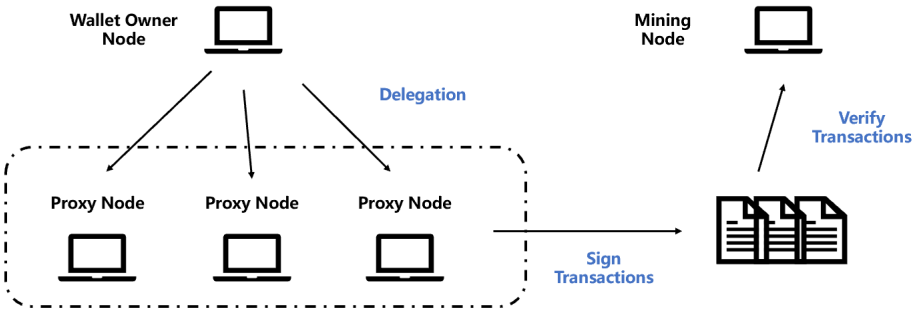

Figure 2. Proxy signature in blockchain.

In the Internet of Things environment, data authentication is of great significance [24,42,43]. Failure to perform integrity verification and authentication of data will lead to serious consequences. However, some edge nodes often have the problem of insufficient resource efficiency. Therefore, it is urgent to use a signature scheme that consumes fewer storage resources in the Internet of Things environment. Our proposed IB-PSSMR scheme can be used in future quantum computing environments in the Internet of Things scenario. For example, in the Internet of Things environment, an organization has many devices, one of which is the main device, and the other devices are also under the organization. At the same time, they share an identity. The proxy signature scheme can be used to authorize the affiliated devices. The traffic sent from the organization is the same identity. As shown in Figure 3, in the Internet of Things, the master device in the group can authorize the slave device by proxy. After the traffic sent by the slave device is signed by the proxy, it can be authenticated by other groups, and it can be attributed to the traffic of the same organization. Similarly, in this process, we need to control the size of the signature within a reasonable range, otherwise it will cause congestion to the traffic of the Internet of Things. The IB-PSSMR scheme can be used as an alternative to the post-quantum scheme in this Internet of Things environment to enhance data authentication.

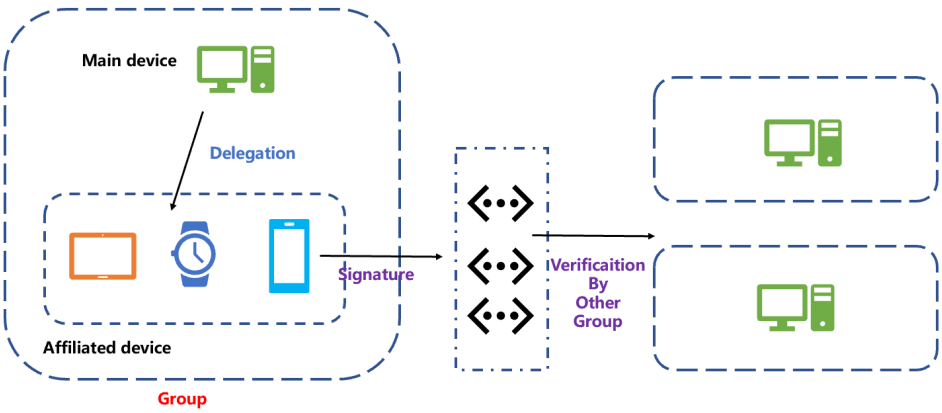

Figure 3. Proxy signature in IOT.

8. Conclusions

Bandwidth is more precious than gold, especially in resource-constrained environments. In the era of quantum computing, it is necessary for us to construct an efficient proxy signature that is quantum safe. Because there are many post quantum schemes that use heavy computation and their signature size is not compact. The lattice- based architecture is the most attractive. In this paper, we construct an efficient identity-based

proxy signature scheme with message recovery (IB-PSSMR) over the NTRU lattice under the standard Gentry–Peikert–Vaikuntanathan (GPV) framework [44]. In spite of the well-studied security proof, our scheme also benefits the excellent computation performance in NTRU lattice and can achieve the message recovery function in the sign phrase. We also give a formal security proof of our proposed scheme, and the efficiency analysis is compared with some related proxy signature construction. In the future, we will continue to improve the usability of our scheme and survey the concrete application scenario of our scheme.

Author Contributions: Methodology, F.W.; Investigation, F.W.; Writing—original draft, F.W.; Writing—review & editing, B.Z.; Supervision, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (2022YFB2703400).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors thank anonymous reviewers and editors for their hard work.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Mambo, M.; Usuda, K.; Okamoto, E. Proxy Signatures: Delegation of the Power to Sign Messages. *IEICE Trans. Fundam. A* **1996**, *79*, 1338–1354.
2. Yang, X.; Gao, G.; Li, Y.; Wang, C. On-line/off-line threshold proxy re-signature scheme through the simulation approach. *Appl. Math. Inf. Sci.* **2015**, *9*, 3251–3261.
3. Kumar, R.; Verma, H.K.; Dhir, R. Analysis and Design of Protocol for Enhanced Threshold Proxy Signature Scheme Based on RSA for Known Signers. *Wirel. Pers. Commun.* **2015**, *80*, 1281–1345. [CrossRef]
4. Shor, P. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium On Foundations Of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp.124–134
5. Tang, S.; Xu, L. *Towards Provably Secure Proxy Signature Scheme Based on Isomorphisms of Polynomials*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 91–97.
6. Yang, C.; Qiu, P.; Zheng, S.; Wang, L. An Efficient Lattice-Based Proxy Signature Scheme without Trapdoor. In Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Adelaide, Australia, 25 February 2016; pp. 189–194.
7. Chen, Y.Z.; Liu, Y.; Wen, X.J. A quantum proxy weak blind signature scheme. *Chin. J. Quantum Electron.* **2011**, *54*, 1325–1333.
8. Zhang, L.; Ma, Y. A Lattice-Based Identity-Based Proxy Blind Signature Scheme in the Standard Model. *Math. Probl. Eng.* **2014**, *2014*, 307637. [CrossRef]
9. Wang, T.Y.; Wei, Z.L. Analysis of Forgery Attack on One-Time Proxy Signature and the Improvement. *Int. J. Theor. Phys.* **2015**, *55*, 1–3. [CrossRef]
10. Micciancio, D.; Regev, O. Worst-Case to Average-Case Reductions Based on Gaussian Measures. In Proceedings of the IEEE Symposium on Foundations of Computer Science, Philadelphia, PA, USA, 18–21 October 2014; pp. 372–381.
11. Simoens, P.; Vankeirsbilck, B.; Deboosere, L.; Ali, F.A.; Turck, F.D.; Dhoedt, B.; Demeester, P. Upstream bandwidth optimization of thin client protocols through latency-aware adaptive user event buffering. *Int. J. Commun. Syst.* **2011**, *24*, 666–690. [CrossRef]
12. Liu, C.X.; Liu, Y.; Zhang, Z.J.; Cheng, Z.Y. High energy-efficient and privacy-preserving secure data aggregation for wireless sensor networks. *Int. J. Commun. Syst.* **2013**, *26*, 380–394. [CrossRef]
13. Nyberg, K.; Rueppel, R.A. A new signature scheme based on the DSA giving message recovery. In Proceedings of the CCS '93: Proceedings of the ACM Conference on Computer and Communications Security, Fairfax, VA, USA, 3–5 November 1993; pp. 58–61.
14. Lyubashevsky, V. Lattice signatures without trapdoors. In Proceedings of the Advances In Cryptology–EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 15–19 April 2012; pp. 738–755
15. Tian, M.; Huang, L. Lattice-based message recovery signature schemes. *Int. J. Electron. Secur. Digit. Forensics* **2013**, *5*, 257–269. [CrossRef]

16. Wu, F.; Yao, W.; Zhang, X.; Zheng, Z. An Efficient Lattice-Based Proxy Signature with Message Recovery. In Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Guangzhou, China, 12–15 December 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 321–331.
17. Lu, X.; Wen, Q.; Yin, W.; Liang, K.; Jin, Z.; Panaousis, E.; Chen, J. Quantum-Resistant Identity-Based Signature with Message Recovery and Proxy Delegation. *Symmetry* **2019**, *11*, 272. [CrossRef]
18. Feng, Y.; Zhou, J.; Li, J.; Zhao, W.; Shi, J.; Shi, R.; Li, W. SKC-CCCO: An encryption algorithm for quantum group signature. *Quantum Inf. Process.* **2022**, *21*, 328. [CrossRef]
19. Lu, D.; Li, Z.; Yu, J.; Han, Z. A verifiable arbitrated quantum signature scheme based on controlled quantum teleportation. *Entropy* **2022**, *24*, 111. [CrossRef] [PubMed]
20. Chen, J.J.; You, F.C.; Li, Z.Z. Quantum multi-proxy blind signature based on cluster state. *Quantum Inf. Process.* **2022**, *21*, 104. [CrossRef]
21. Feng, Y.; Shi, R.; Shi, J.; Zhao, W.; Lu, Y.; Tang, Y. Arbitrated quantum signature protocol with boson sampling-based random unitary encryption. *J. Phys. A Math. Theor.* **2020**, *53*, 135301. [CrossRef]
22. Feng, Y.; Shi, R.; Shi, J.; Zhou, J.; Guo, Y. Arbitrated quantum signature scheme with quantum walk-based teleportation. *Quantum Inf. Process.* **2019**, *18*, 154. [CrossRef]
23. Fang, W.; Chen, W.; Zhang, W.; Pei, J.; Gao, W.; Wang, G. Digital signature scheme for information non-repudiation in blockchain: A state of the art review. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 2643546. [CrossRef]
24. Verma, G.K.; Singh, B.; Kumar, N.; Obaidat, M.S.; He, D.; Singh, H. An efficient and provable certificate-based proxy signature scheme for IIoT environment. *Inf. Sci.* **2020**, *518*, 142–156. [CrossRef]
25. Zhang, J.; Bai, W.; Wang, Y. Non-interactive ID-based proxy re-signature scheme for IoT based on mobile edge computing. *IEEE Access* **2019**, *7*, 37865–37875. [CrossRef]
26. Verma, G.K.; Singh, B.; Kumar, N.; He, D. CB-PS: An efficient short-certificate-based proxy signature scheme for UAVs. *IEEE Syst. J.* **2019**, *14*, 621–632. [CrossRef]
27. Xie, J.; Hu, Y.p.; Gao, J.t.; Gao, W. Efficient identity-based signature over NTRU lattice. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 135–142. [CrossRef]
28. Lyubashevsky, V. Lattice Signatures without Trapdoors. In Proceedings of the 31st Annual international conference on Theory and Applications of Cryptographic Techniques, Athens, Greece, 26–30 May 2013; pp. 738–755.
29. Barr, K.C. Energy-aware lossless data compression. *Acm Trans. Comput. Syst.* **2006**, *24*, 250–291. [CrossRef]
30. Peikert, C. An efficient and parallel Gaussian sampler for lattices. In Proceedings of the Annual Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 80–97.
31. Sinha Roy, S.; Vercauteren, F.; Verbauwhede, I. High precision discrete Gaussian sampling on FPGAs. In Proceedings of the International Conference on Selected Areas in Cryptography, Burnaby, BC, Canada, 14–16 August 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 383–401.
32. Ducas, L.; Nguyen, P.Q. Faster Gaussian lattice sampling using lazy floating-point arithmetic. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, 1–5 December 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 415–432.
33. Buchmann, J.; Cabarcas, D.; Göpfert, F.; Hülsing, A.; Weiden, P. Discrete Ziggurat: A time-memory trade-off for sampling from a Gaussian distribution over the integers. In Proceedings of the International Conference on Selected Areas in Cryptography, Burnaby, BC, Canada, 14–16 August 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 402–417.
34. Cai, J.Y.; Nerurkar, A. Approximating the SVP to within a factor (1-1/dim/sup/spl epsiv//) is NP-hard under randomized conditions. In Proceedings of the Thirteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat. No. 98CB36247), Buffalo, NY, USA, 18 June 1998; IEEE: New York, NY, USA, 1998; pp. 46–55.
35. Zhang, J.; Yu, Y. Short computational Diffie–Hellman-based proxy signature scheme in the standard model. *Int. J. Commun. Syst.* **2014**, *27*, 1894–1907. [CrossRef]
36. Apostol, T.M. Resultants of cyclotomic polynomials. *Proc. Am. Math. Soc.* **1970**, *24*, 457–462. [CrossRef]
37. Bellare, M.; Neven, G. Multi-signatures in the plain public-Key model and a general forking lemma. In Proceedings of the ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, 30 October–3 November 2006; pp. 390–399.
38. Agrawal, S.; Boyen, X. Identity-Based Encryption from Lattices in the Standard Model. *Manuscript* **2009**, *3*. Available online: http://www.cs.stanford.edu/xb/ab09/ (accessed on 5 February 2023).
39. Wu, F.; Yao, W.; Zhang, X.; Wang, W.; Zheng, Z. Identity-based proxy signature over NTRU lattice. *Int. J. Commun. Syst.* **2019**, *32*, e3867. [CrossRef]
40. Ducas, L.; Lyubashevsky, V.; Prest, T. Efficient identity-based encryption over NTRU lattices. In Proceedings of the Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Part II 20, Kaoshiung, Taiwan, 7–11 December 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 22–41.
41. Wang, Y.; Qiu, W.; Dong, L.; Zhou, W.; Pei, Y.; Yang, L.; Nian, H.; Lin, Z. Proxy signature-based management model of sharing energy storage in blockchain environment. *Appl. Sci.* **2020**, *10*, 7502. [CrossRef]

42. Qiao, Z.; Zhou, Y.; Yang, B.; Zhang, M.; Wang, T.; Xia, Z. Secure and efficient certificate-based proxy signature schemes for industrial internet of things. *IEEE Syst. J.* **2021**, *16*, 4719–4730. [CrossRef]
43. Hussain, S.; Ullah, I.; Khattak, H.; Khan, M.A.; Chen, C.M.; Kumari, S. A lightweight and provable secure identity-based generalized proxy signcryption (IBGPS) scheme for Industrial Internet of Things (IIoT). *J. Inf. Secur. Appl.* **2021**, *58*, 102625. [CrossRef]
44. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, Victoria, BC, Canada, 17–20 May 2008; pp. 197–206.