Research Article
An Efficient Identity-Based Proxy Blind Signature for Semioffline Services

Hongfei Zhu,1 Yu-an Tan,1 Liehuang Zhu,1 Quanxin Zhang,1 and Yuanzhang Li1,2

1School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
2Research Center of Massive Language Information Processing and Cloud Computing Application, Beijing 100081, China

Correspondence should be addressed to Yuanzhang Li; popular@bit.edu.cn

Received 6 February 2018; Accepted 12 March 2018; Published 9 May 2018

Copyright © 2018 Hongfei Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fog computing extends the cloud computing to the network edge and allows deploying a new type of semioffline services, which can provide real-time transactions between two entities, while the central cloud server is offline and network edge devices are online. For an e-payment system and e-voting with such feature, proxy blind signature is a cornerstone to protect users’ privacy. However, the signature based on number theorem, such as hard mathematical problems on factoring problem, discrete logarithm problem, and bilinear pairings, cannot defeat quantum computers attack. Meanwhile, these schemes need to depend on complex public key infrastructure. Thus, we construct an identity-based proxy blind signature scheme based on number theorem research unit lattice, which can defeat quantum computers attack and does not need to depend on public key infrastructure. The security of the proposed scheme is dependent on Ring-Small Integer Solution problem over number theorem research unit lattice. The proposed scheme meets the properties of blind signature and proxy signature. Then we compare the proposed scheme with other existing proxy blind signature schemes; the result shows that the proposed scheme outperforms ZM scheme except in proxy signer’s signature size and can be more secure than TA scheme and MMHP scheme.

1. Introduction

Fog computing was initially introduced by Cisco, which can overcome cloud computing’s disadvantages, such as non-real-time service and long delay [1–3]. More specifically, fog computing adds a new layer between cloud server and terminal user [4, 5]; that is, fog servers can be access point, base station, router, or mobile equipment [6–9]. Thus, the semioffline e-payment system can be deployed by utilizing the advantages of fog computing model [10, 11].

In order to defend user’s privacy in offline e-payment system, blind signature (BS) is crucial for that it never permits signer to sign on a plaintext before knowing its content [12]. Therefore, BS can protect user privacy during the transactions [13] instead of encrypting the data and searching on the ciphertext [14]. However, this system is deployed in real environment; it will use distributed architecture [15]. The original signer should authorize an agent to sign for himself. Then a proxy signature (PS) should be used in e-payment system, since proxy signer can satisfy this requirement [16].

Combining those two types of schemes together, a new proxy blind signature (PBS) was proposed, which meets the properties of those two signature schemes. After that, many PBS schemes were constructed by scholars.

However, most of the PBS schemes are based on number theory, such as discrete logarithm problem (DLP) and bilinear pairings. These schemes are considered to be insecure to resist the quantum computer attack. Therefore, the e-payment and e-voting systems in the cloud still face the threat from quantum computer attack [17]. Meanwhile, these schemes need to rely on complex public key infrastructure (PKI) [18, 19]. In conclusion, these schemes based on number theorem cannot defeat the quantum computers attack according to the recent research results.

Therefore, the lattice-based PBS schemes become one alternative solution, since they are sufficient enough and able to resist quantum computer attack [20, 21]. Besides, if lattice-based PBS schemes can combine with identity-based cryptography (called IDPBS), they can overcome the shortcomings of traditional PBS schemes, such as relying on...
complex PKI [22]. Meanwhile, they can transfer less data than biological recognition methods during the transactions [23, 24].

Zhu et al. presented a new lattice-based BS [20], which can be secure enough for cloud services. However, this scheme has to be combined with proxy signature in practice. Combining BS scheme and IDPBS scheme, we initially present an IDPBS on number theory research unit lattice (IDPBS-NTRU), which can defeat quantum computer attack.

(1) Inspired by [25], a new IDPBS-NTRU scheme is proposed based on NTRU lattice, which can make semioffline e-payment and e-voting systems deployed in fog computing model secure enough to resist quantum computer attack.

(2) The proposed IDPBS-NTRU scheme is proven to be secure. That is, the proposed scheme is correct, blind, unforgeable, verifiable, strong identifiable, strong undeniable, and key-dependent.

(3) The proposed IDPBS-NTRU is compared with other IDPBS schemes in terms of performances. At last, Section 5 draws the conclusions.

2. Related Works

IDPBS Schemes Based on DLP. In 2011, Beura et al. proposed a new proxy blind signature based on DLP; their scheme satisfies the properties of blind signature and proxy signature. This scheme is more secure and efficient than factoring signature schemes [26]. To improve the efficiency, Tan et al. introduced a couple of PBS schemes; both of them were constructed on Schnorr blind signature. However, Sun et al. pointed that both of them were not unforgeable and unlinkable [27]. However, in 2014, Wang and Liao proved that the schemes proposed by Oo et al. and Beura et al. did not satisfy unlinkability [28]. In 2013, Tan proposed a PBS based on DLP, which did not depend on PKI [29] and was proven to be secure in the random oracle [30]. However, most of these schemes are dependent on PKI and are not strictly proven to be secure.

IDPBS Schemes Based on Bilinear Pairings. In 2003, Zhang et al. proposed a new proxy blind signature based on bilinear pairings, which satisfies distinguishingability, verifiability, strong nonforgeability, strong identifiability, strong nondeniability, and prevention of misuse. Meanwhile, this scheme did not depend on public key infrastructure (PKI) [31]. Later, Li et al. introduced a new PBS, which was also constructed on bilinear pairings; it was independent of PKI [32]. However, these schemes are inefficient and not proven to be secure.

IDPBS Schemes Based on Lattice. In 2014, Zhang and Ma initially proposed a proxy blind signature on lattice; it does not need to depend on PKI; its security is based on short integer solution problem. However, this scheme is still inefficient [33].

3. Preliminaries

In the beginning, we will define the denotations that will be used all over the paper in Denotations.

3.1. NTRU Lattice and Rejection Sampling on Lattice

Definition 1 (NTRU lattice). The notations are defined as \( f, g \in \mathbb{R} \) and \( h = gf^{-1} \mod q \); after that, the NTRU lattice can be defined as \( \mathcal{L}_{h,q} = \{ u, v \in \mathbb{R}^2 : u + vh = 0 \mod q \} \). That is, \( \mathcal{L}_{h,q} \) is on behalf of a \( \mathbb{R}^{2N} \) full-rank lattice whose basis is \( \begin{pmatrix} -T_{d,h} & 1 \end{pmatrix} \in \mathbb{R}^{2N} \). \( I_n \) denotes a unit matrix, \( O_{k} \) denotes a null matrix, and \( T_N(h) \) denotes an anticirculant matrix

\[
\begin{pmatrix}
-h_0 & h_1 & -h_{N-1} & \\
-h_1 & h_2 & -h_{N-2} & \\
\vdots & \vdots & \ddots & \\
-h_{N-1} & h_N & -h_1 & \\
-h_N & h_0 & -h_2 & \\
\end{pmatrix}.
\]

Definition 2 (R-SIS\( ^{k,1,2,\beta} \) on NTRU lattice). Small \( f \) and \( g \) can be sampled from \( D_{Z^N,\rho} \), \( (f,g \mod q \in \mathbb{R}^2) \); then \( \mathcal{L}_{h,q} = (h,1) \in \mathbb{R}^{2N} \) and \( h = gf^{-1} \) can be obtained by using Algorithm 3 in [25]. Therefore, R-SIS on NTRU means finding \( z_1, z_2 \) satisfying \( \mathcal{L}_{h,q}(z_1, z_2)^T = 0 \mod q \) and \( \|(z_1, z_2)\| \leq \beta \).

Theorem 3 (rejection sampling theorem). \( V \) denotes one subset of \( \mathbb{Z}^m \), the norms of \( V \)'s elements are less than constant \( T, \sigma = \omega(T \sqrt{\log m}) \in \mathbb{R}, (M \text{ is invariable}), \) and \( h : V \rightarrow \mathbb{R} \) is a probability distribution. Two algorithms are as follows: One is

(1)

\[
\begin{align*}
\text{v} & \leftarrow h; \\
\text{w} & \leftarrow D_{\text{v},\sigma}; \\
\text{get}(w,v) \text{ with probability } & \min\left(\frac{D_{\text{v}}(w)}{MD_{\text{v},\sigma}(w)}, 1\right).
\end{align*}
\]

The other is

(2)

\[
\begin{align*}
\text{v} & \leftarrow h; \\
\text{w} & \leftarrow D_{\text{v}}; \\
\text{get}(w,v) \text{ with probability } & \frac{1}{M}.
\end{align*}
\]

Then the distribution of first algorithm will not exceed the second one's statistical distance \( 2^{-\omega(\log N)} \mod M \). Moreover, The first one will export something with probability at least \( (1 - 2^{-\omega(\log N)}) \mod M \).

3.2. The Definitions of IDPBS Scheme. An IDPBS consists of seven algorithms (ST\( _c \), KE\( _c \), DG\( _c \), BS\( _c \), PS\( _c \), UB\( _c \), PV\( _c \))
Blindness. The details can be seen in [20, 33, 40–42]. TTP will execute $ST_i(1^N)$ to produce public parameters and keys [29, 38, 39]. The formal definition is presented as follows (Algorithm 1):

(i) $ST_i(1^N)$ outputs $\text{params}$ and $\text{mpk} = (\text{msk}, \text{mpk})$.

(ii) $KE_i(\text{params}, \text{msk}, \text{id}_i)$ outputs $\text{id}_{sk_i}$ for $\emptyset$ and $\pro$ ($i = \emptyset$ or $\pro$).

(iii) $DG_i(\text{id}_{id_i}, \text{sk}_{id_i}, \omega)$ outputs $\psi$ for $\pro$.

(iv) Proxy blind signature: $\mathcal{U}$ interacts with $\pro$ according to the following protocol:

1. $BS_i(m, f)$: $\mathcal{U}$ blinds $m$ to $m'$ by using $f$ and then sends $m'$ to $\pro$.

2. $PS_i(m', \text{id}_p, \text{sk}_p, \text{id}_{sp})$: $\pro$ signs on $m'$ using $\text{sk}_p$ and sends the signature $\sigma'$ to $\mathcal{U}$.

3. $UB(\sigma', f)$: $\mathcal{U}$ unblinds $\sigma'$ by using $f$ and outputs the blind signature $\sigma$.

(v) $PV_i(\text{id}_{id_i}, \text{id}_p, \omega, m, \psi, \sigma)$: if $\psi, \sigma$ are valid, the algorithm outputs true. Otherwise it outputs false.

An IDPBS scheme should meet the following six properties. The details can be seen in [20, 33, 40–42].

1. **Blindness.** $P^*$ are denoted as an adversary who can control the proxy signer. $P^*$ chooses two messages $m_0$ and $m_1$. Then a random bit $i \in \{0, 1\}$ is chosen in the game. $m_0$ and $m_1$ are randomly denoted as $m_i$ and $m_{1-i}$, respectively, used as two user’s inputs. After that, $P^*$ will adaptively and parallelly interact with two honest users according to the signature protocol. Finally, two users output $\sigma_i$ and $\sigma_{1-i}$, respectively. Then $\sigma_i$ and $\sigma_{1-i}$ ordered by $m_i$ and $m_{1-i}$ are delivered to $P^*$; after that, $P^*$ will output $i \in \{0, 1\}$.

2. **One More Unforgeability.** $\pro$ can generate a legal proxy $\sigma$ instead of $\emptyset$. However, $\emptyset$ and all the other entities fail to generate a legal signature. The game is presented as follows [33]: $Adv_{V_{\pro}}$. the advantage of $U^*$, is denoted as success probability in $Algorithm$. If no adversary can win $Algorithm$ at minimum with negligible probability $\eta$ in time $t$, then it satisfies one more unforgeability [31].

(3) **Verifiability.** $\mathcal{V}$ can check whether $\sigma$ is delegated by $\emptyset$.

(4) **Strong Identifiability.** Any $\mathcal{V}$ can determine $\pro$’s identity once he receives the proxy signature tuple.

(5) **Strong Undeniability.** $\pro$ cannot refuse to admit it once he creates the proxy signature $\sigma$.

(6) **Key Dependence.** $\pro$ can sign on a message if and only if he has the authorization from $\emptyset$.

**4. Proposed IDPBS-NTRU Scheme**

Here, we introduce a novel IDPBS-NTRU $\epsilon = (ST_i, KE_i, DG_i, BS_i, PS_i, U B_i, P V_i)$, which can be seen in $Algorithm 4$. The details are as follows.

$1. ST_i(1^N), q = \text{Poly}(N), \epsilon \in (0, \ln N/\ln q)$, and $s = \tilde{\Omega}(N^{3/2})$. If $N > 2$, $\sigma = N \sqrt{\ln \left(8Nqq\right)}^{1/2-\epsilon}, q^{1/2-\epsilon} = \tilde{\Omega}(\sqrt{N})$. If $N = 2$, $\sigma = \sqrt{\ln \left(8Nq\right)}^{1/2-\epsilon}, q^{1/2-\epsilon} = \tilde{\Omega}(N)$. $\text{msk}$ and $\text{mpk}$ can be obtained as below [25].

The algorithm takes samples $f$ and $g$ from $D_{2^N}$. Here, $f, g \mod q \notin R_q^*$; if $\|f\|$, $\|g\| \leq \sigma \sqrt{N}$, and $(f, g) \in \mathbb{R}$. After that, the algorithm can get $F_i, G_i \in \mathbb{R}$ according to the equation $f G_i - g F_i = 1$. Given $F_i = q F_i$ and $G_i = q G_i$, $(F_i, G_i)$ can be obtained from $(f, g), (xf, xg), \ldots, (x^{N-1}f, x^{N-1}g)$ according to Babai algorithm [43]. Then there exists $(F_i, G_i) = (F_q, G_q) - k(f, g)$. If $\|F_i, G_i\| \leq N_\sigma$, then the algorithm outputs system parameters $\text{paras} = (q, \epsilon, s)$; the master private-key $\text{msk}$ and master public-key $\text{mpk}$ are as follows:
\[
\text{msk} = B = \begin{pmatrix} T(f) & T(g) \\ T(F) & T(G) \end{pmatrix},
\]
\[
mpk = h = gf^{-1} \in \mathbb{R}_q^*.
\]

(2) KE\(_e\) \((\text{paras, id}_i, \text{msk})\): The algorithm executes (4) to get an \(N\)-dimension matrix \(t\); then the algorithm executes (6) and outputs \(sk_{id_i}\) according to corresponding \(id_i\) \((i = \emptyset, \mathcal{P})\) [25].

\[
t \leftarrow H_1 \left( \text{id}_i \right),
\]

\[
\left( s_i, s_{i_u} \right) \leftarrow \left( (t, 0) - \text{Gaussian} \left( \text{msk}, \sigma, (t, 0) \right) \right),
\]

\((iii)\) The algorithm executes (7) and (8) to generate valid delegations \(\sigma_1\) and \(\sigma_2\). Here, the algorithm uses the rejection sampling theorem to keep the delegation independent on \(\mathcal{O}'s\) secret keys \(s_{\emptyset_1}\) and \(s_{\emptyset_2}\).

(4) \(BS_{\emptyset}\). \(m\) is a plaintext. \(\mathcal{U}\) will execute this algorithm to generate a blind message.

\((i)\) The algorithm will randomly select \(y_1, y_2 \in D_{2^{N_q}}\) at random.

\((ii)\) The algorithm executes (6) to get the \(N\)-dimension matrix \(u\).

\((iii)\) The algorithm executes (7) and (8) to generate valid delegations \(\sigma_1\), \(\sigma_2\), \(\sigma_3\), \(\sigma_4\). Here, the algorithm uses the rejection sampling theorem to keep the delegation independent on \(\mathcal{O}'s\) secret keys \(s_{\emptyset_1}\) and \(s_{\emptyset_2}\).

(5) \(PS_{\emptyset}\). The proxy signer \(\mathcal{P}\) will execute this algorithm to sign the blinded message.

\((i)\) The algorithm validates whether (11) and (12) are true. If either is false, \(\mathcal{P}\) aborts the algorithm. Otherwise, it continues.

\((ii)\) The algorithm will execute (13) and (14). Here, the rejection sample theory is used to keep the proxy signatures \(\sigma_1\) and \(\sigma_4\) independent on \(\mathcal{P}'s\) secret keys \(s_{\emptyset_1}\) and \(s_{\emptyset_2}\).

(6) \(UB_{\mathcal{U}}\). \(\mathcal{U}\) will execute the algorithm to unblind the proxy signature.

\((i)\) The algorithm executes (15) to unblind the proxy signature tuple.

\((ii)\) \(\mathcal{U}\) outputs the signature tuple \((\sigma_1, m, \sigma_2, \omega, u, \sigma_3, \sigma_4, e)\).

\[
\sigma_3 = \sigma_3^* + \alpha,
\]

\[
\sigma_4 = \sigma_4^* + \gamma.
\]

(7) \(PV_\mathcal{V}(\sigma_1, m, \sigma_2, \omega, u, \sigma_3, \sigma_4, e, id_{\emptyset_1}, id_{\emptyset_2})\). \(\mathcal{V}\) will execute this algorithm to validate whether the signature tuple satisfies (16). If all the equations mentioned above are true, the signatures are valid. Otherwise, they are invalid.

\[
\| (\sigma_1, \sigma_2) \| \leq 2s \sqrt{2N},
\]

\[
H_2 \left( h\sigma_2 + \sigma_1 - H(\text{id}_{\emptyset_2}) u, \omega \right) = u,
\]

\[
\| (\sigma_3, \sigma_4) \| \leq 2s \sqrt{2N},
\]

\[
H_3 \left( h\sigma_4 + \sigma_3 - H(\text{id}_{\emptyset_2}) e, m \right) = e.
\]

\section{5. Security and Performance Comparison}

5.1. Security

\(1)\)

\textbf{Theorem 4} (correctness). The IDPBS-NTRU scheme is correct.

\textit{Proof.} According to the construction of our IDPBS scheme, we can get

\[
h \cdot \sigma_2 + \sigma_1 - H(\text{id}_{\emptyset_2}) \cdot u
\]

\[
= h \left( y_1 + s_{\emptyset_1} u \right) + y_1 + s_{\emptyset_1} u - H(\text{id}_{\emptyset_2}) \cdot u
\]

\[
= h \left( y_1 + hy_2 \right)
\]

Therefore, \(H_2 \left( (h \cdot \sigma_2 + \sigma_1 - H(\text{id}_{\emptyset_2}) \cdot u, \omega \right) = u\).

\[
h\sigma_4 + \sigma_3 - H(\text{id}_{\emptyset_2}) e
\]

\[
= h \left( \sigma_4^* + \gamma \right) + \sigma_3^* + \alpha - H(\text{id}_{\emptyset_2}) \cdot e
\]

\[
= h \left( y_4 + s_{\emptyset_2} \epsilon^* + \gamma \right) + y_3 + \alpha + s_{\emptyset_2} \epsilon^* - H(\text{id}_{\emptyset_2}) \cdot e
\]

\[
\cdot e = y_3 + hy_4 + hy + \alpha - \alpha H(\text{id}_{\emptyset_2})
\]

Thus, \(H_3(h\sigma_4 + \sigma_3 - H(\text{id}_{\emptyset_2}) e, m) = e\).

\[\square\]
which is kept secret from respectively. Finally, these two honest users output \( \sigma_m \) will be sent to \( P^* \). 

Theorem 5 (blindness). The IDPBS-NTRU scheme satisfies blindness.

Proof. As shown in Algorithm 2, a random bit \( i \in \{0, 1\} \) which is kept secret from \( P^* \). Then \( P^* \) chooses two messages \( m_0 \) and \( m_1 \) and \( m_i \) are the inputs of two honest users, respectively. \( P^* \) adaptively and parallely interacts with two honest users, respectively. Finally, these two honest users output \( \sigma_i \) and \( \sigma_{1-i} \), respectively. The sequence \( \sigma_i \) and \( \sigma_{1-i} \), ordered by \( m_i \) and \( m_{1-i} \) will be sent to \( P^* \). \( P^* \) will output a bit \( i' \in \{0, 1\} \).

In the process of signature protocol, all intermediate results do not depend on \( m_i \); thus it is enough to analyze \( e^* \), \( y_3, y_4, \sigma_1, \sigma_2, u, \omega, \sigma_1^*, \) and \( \sigma_2^* \).

To \( e^* \), the statistical distance is presented as follows:

\[
\Delta(e^*_i, e_{i-1}^*) = \frac{1}{2} \sum_{e^* \in \mathcal{P}_\mathcal{N}_Y} \left| \Pr(e^*_i = e^*) - \Pr(e^*_{i-1} = e^*) \right|.
\]

Since \( \alpha \) is chosen at random, next we obtain the equations 
\( \Pr(e^*_i = e^*) = 1/2 \) and \( \Pr(e^*_{i-1} = e^*) = 1/2 \). Therefore, we obtain \( \Delta(e^*_i, e_{i-1}^*) = 0 \).

In the same way, we can obtain \( \Delta(y_3^*, y_4^*) = 0 \), \( \Delta(y_3, y_4^{-1}) = 0 \), \( \Delta(\sigma_1^*, \sigma_1^{-1}) = 0 \), \( \Delta(\sigma_2^*, \sigma_2^{-1}) = 0 \), \( \Delta(u^*, \omega^{-1}) = 0 \), \( \Delta(u^*, \omega^{-1}) = 0 \), \( \Delta(\sigma_3^*, \sigma_3^{-1}) = 0 \), and \( \Delta(\sigma_4^*, \sigma_4^{-1}) = 0 \). Therefore, \( \delta^* \) cannot distinguish \( m \) among \( e^*, y_3, y_4, \sigma_1, \sigma_2, u, \omega, \sigma_1^* \), and \( \sigma_2^* \); that is, \( \delta^* \) can win this experiment with probability \( 1/2 \). Thus, the theorem is proven.

We denote \( \delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6 \) as the cost functions of \( H_1, H_2, H_3 \) hash oracle, extract Oracle, DG oracle, and signature oracle, respectively, \( \eta \) as a nonnegligible probability, \( \Theta \) as a polynomial time algorithm, and \( \Gamma \) as a polynomial time forger.

(3)

Theorem 6 (one more unforgeability). If \( \Gamma \) is able to generate a legal IDPBS signature with \( \eta \) in \( t \) successfully, after at most \( t_1, t_2, t_3, t_4, t_5, t_6 \) times queries, respectively, to \( H_1, H_2, H_3, H_4, H_5 \) hash, extract, DG, and blind signature oracles, then R-SIS \( \mathbb{S}_{k, l, b}^\delta \) can be solved by \( \Theta \) with probability at least \( \eta' = \left(1 - 2^{-\log N} \eta\right)^t \) in time \( t' = t + t_1^5 + t_2^6 + t_3^4 + t_4^5 + t_5^2 + t_6^2 \).

Proof. We suppose that \( \Gamma \) is able to generate an IDPBS signature successfully with \( \eta \); we are able to construct \( \Theta \) to calculate R-SIS’s solution. The interaction environment can be simulated as follows.

Algorithm 3: Exp_{\mathcal{N}_Y}^{\text{mof}}(n).

\[
\text{Setup.} \Theta \text{ selects } h \in \mathbb{R}_k^* \text{ and } H_1, H_2, H_3 \text{ at random. Next } \Theta \text{ calculates and delivers } \text{paras} = \{h, H_1, H_2, e, q, s\} \to \Gamma.
\]

Queries on \( H_1 \) Oracle. To reply to \( H_1 \) oracle’s query, \( \Theta \) creates one null list \( L_1 \). Once \( \Theta \) obtains one \( id_i \), \( \Theta \) will query \( H_1 \). If there is a \( t_i \) consistent with the query, \( \Theta \) will return \( t_i \). Otherwise, \( \Theta \) will select a random \( t_i \). At last, \( \Theta \) will return \( t_i \) to \( \Gamma \) and save (\( id_i, t_i \)) to \( L_1 \).

Queries on \( H_2 \) Oracle. To reply to \( H_2 \)’s queries, \( \Theta \) creates a list \( L_2 \) is initialized null. When \( \Theta \) receives an \( (y_1, y_2, u) \), \( \Theta \) will query \( H_2 \). If there is one corresponding value \( t \), \( \Theta \) will return \( u \). Else \( \Theta \) will choose one \( u \) at random. Finally, \( \Theta \) will return \( u \) to \( \Gamma \) and save \( (y_1, y_2, u, w) \) to \( L_2 \).

KE Oracle Queries. When \( \Theta \) queries a private key related to one \( id_i \), \( \Gamma \) will recover the corresponding \( (id_i, t_i) \) from \( L_1 \). Next \( \Gamma \) will run \( sk_{id} = (s_1, s_2) \to [(t_0) - \text{Gaussian}(s, \sigma, (t_0))] \to \Gamma \) will return \( id_i, t_i, sk_{id} \) to \( \Theta \) and save the tuple to \( L_3 \).

DG Oracle Queries. When \( \Theta \) requests the delegation queries, \( \Theta \) will verify whether \( id_i \) has been queried for \( H_1 \) or \( KE_x \) oracle. If it has been queried, \( \Theta \) will obtain \( (id_i, t_i, sk_{id}) \) from \( L_3 \). Else \( \Theta \) will simulate KE_x oracle and get a new private key. Next \( \Theta \) will execute \( \sigma_1 = y_1 + s_1, u \) and \( \sigma_2 = y_2 + s_1, u \) to get a legal delegation signature \( (u, \sigma_1, \sigma_2) \) and save \( (y_1, y_2, m_i, \sigma_1, \sigma_2) \) to \( L_4 \).
compute for the first time.

$\text{Signature Oracle Queries.}$ $\Gamma$ queries the signing oracle for $y_{i1}, y_{i2}, \beta, id_{i},$ and $m_{i}$. $\Theta$ will verify whether $id_{i}$ has been queried for $H_{1}$ or $KE_{1}$ oracle. If it has been queried, $\Theta$ will obtain $(id_{i}, t_{i}, sk_{i, d})$ from $L_{i}$. Else $\Theta$ will simulate the $EX_{c}$ oracle and get a new private key. Next $\Theta$ will execute $DG$ queries and then obtain $(y_{i}, y_{i1}, y_{i2}, u_{i}, s_{i, i}, 1, \sigma_{i})$ from $L_{i}$. Then $\Theta$ will execute $\sigma_{i} = y_{i1} + s_{i, i} e_{i} + s_{i, i} e_{i}$ to get a valid signature $(m_{i}, u_{i}, s_{i, i}, 1, \sigma_{i}, e_{i})$ and save $(y_{i1}, y_{i2}, u_{i}, s_{i, i}, 1, \sigma_{i}, e_{i})$ to $L_{i}$.

$\text{Output.}$ At last, $\Gamma$ will output one forged signature $(u_{i}, \omega_{i}, s_{i, i}, 1, m_{i}, \sigma_{i}, 1, \sigma_{i})$ on $u_{i}, m_{i}, id_{\rho},$ and $id_{\rho}$ for the first time. $\Theta$ will rewind $\Gamma$ to the point where $u_{i}$ and $m_{i}$ are queried for $H_{1}$; next $\Gamma$ will get a valid tuple $(u_{i}, \omega_{i}, s_{i, i}, 1, m_{i}, \sigma_{i}, 1, \sigma_{i})$ once again.

Thus, $\Theta$ are able to solve $\text{R-SIS}_{q,1,2,\beta}$ problem. $\Theta$ will compute $\sigma_{i} = y_{i1} + s_{i, i} e_{i}, \sigma_{i} = y_{i1} + s_{i, i} e_{i},$ and $\sigma_{i} + s_{i, i} h - H(id_{\rho}, e_{i})$. Next $\Theta$ will verify whether $\sigma_{i} + s_{i, i} h - H(id_{\rho}, e_{i}) = \sigma_{i} + s_{i, i} h - H(id_{\rho}, e_{i})$ is a solution to $\text{R-SIS}_{q,1,2,\beta}$. Similarly, we can obtain that $(\sigma_{i} - \sigma_{i, i})$ is a solution to $\text{R-SIS}_{q,1,2,\beta}$.

After that, we begin to analyze $\Theta$’s advantages. As mentioned above, $\Theta$ will win this game if $\Gamma$ has already forged a valid $(\sigma_{i} + \omega_{i}, m_{i}, \sigma_{i}, 1, \sigma_{i}, 1, \sigma_{i})$ and $(\sigma_{i}, 1, \sigma_{i}) \neq (\sigma_{i}, 1, \sigma_{i})$ and $(\sigma_{i}, \omega_{i}) \neq (\sigma_{i}, \omega_{i})$. The simulation of the $EX_{c}$ oracle fails if $H_{2}$ causes inconformity. Then $\Theta$ is able to solve $\text{R-SIS}_{q,1,2,\beta}$ with probability at minimum $\eta = ((1 - 2^{-\omega_2N}) \eta)^2$ [25]; here, $\beta = 4s\sqrt{2N}$; it is clear that $t_{i}' = t_{i} + t_{i} + (r_{i, i} + r_{i, i} + r_{i, i}) + r_{i, i} + r_{i, i}$. Therefore, we can prove the theorem.

(4) $\text{Verifiability.}$ Once receiving $(w, m, \sigma_{i}, 1, \sigma_{i}, 1, e_{i}, 1, id_{\rho})$, $V$ will execute $PV_{s}$ to check whether $||\sigma_{i}, \sigma_{i}|| \leq 2s\sqrt{2N}$ and $H_{2}(\sigma_{i} + \sigma_{i}, 1, H(id_{\rho}, u, w)) = u$ are true. If both are true, the proxy signer is delegated by $\Theta$ to sign on $m$.

(5) $\text{Strong Identifiability.}$ After receiving $(w, m, \sigma_{i}, 1, \sigma_{i}, 1, e_{i}, 1, id_{\rho}, \sigma_{i}, 1, e_{i}, 1, id_{\rho}), V$ can confirm $\rho'$s identity in accordance with $id_{\rho}$; thus the IDPBS-NTRU scheme satisfies strong identifiability.

(6) $\text{Strong Undeniability.}$ $\sigma_{i}$ and $\sigma_{i}$ are signed by using $\rho'$s secret keys $s_{\rho}$ and $s_{\rho}$; they will only be known by $\rho'$; thus $\rho'$ cannot refuse his signature once he signed; thus the IDPBS-NTRU scheme satisfies strong undeniability.
(7) Key Dependence. $\sigma_1$ and $\sigma_2$ on warrant $\omega$ are signed by $\theta$’s secret keys $s_{\sigma_1}$ and $s_{\sigma_2}$; they are only known by $\theta$; $\mathcal{P}$ has no legal right to sign on a message before he is authorized by $\theta$; thus the IDPBS-NTRU scheme satisfies key dependence.

5.2. Performance. In this section, we compare the performance of IDPBS-NTRU with other IDPBS schemes. $\lambda$ is written as security parameter, we denote $\theta$’s signature size and signing-key size as OSS and OSK, respectively. $\mathcal{P}$’s signature size and signing key size are denoted as PSS and PSK, respectively. In ZM scheme [33], the parameters satisfy $m \geq 2N \log q$ and $q > \beta \omega \log n$. In TA [34] and MMHP [35] schemes, the security parameter $m$ is equal to $N$.

In Table 1, we compute the signature size and signing key size for $\theta$ and $\mathcal{P}$. It is clear to draw a conclusion that our proxy signer’s OSS, OSK, and PSK are smaller than ZM, TA, and MMHP schemes, our PSS is larger than ZM scheme, our PSS is smaller than TA scheme and MMHP scheme, and our OSS, OSK, PSK, and PSS are larger than TA and MMHP schemes. However, TA scheme and MMHP scheme are based on DLP; they are considered as not secure to resist the quantum computer attack. So our scheme can be more secure than them.

6. Conclusions

In this work, we present an IDPBS-NTRU scheme by using NTRU lattice; this scheme plays an important role in offline e-payment system, which can be deployed in fog computing model. We demonstrate that IDPBS-NTRU is efficient and secure. In addition, our IDPBS-NTRU’s OSS, OSK, and PSK are smaller than ZM scheme and safer than TA and MMHP schemes. The proposed scheme is constructed based on NTRU lattice, which has the advantages of NTRU lattice. In the future, we will continue to construct a partial IDPBS scheme based on lattice.

### Table 1: Performance comparison with other lattice-based IDPBS schemes.

<table>
<thead>
<tr>
<th>Problem</th>
<th>ZM [33]</th>
<th>TA [34]</th>
<th>MMHP [35]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSS</td>
<td>$m^2 \log(\lambda + 1)$</td>
<td>$3m$</td>
<td>$m$</td>
<td>$2N \log(12\sigma) + N (\log \lambda + 1)$</td>
</tr>
<tr>
<td>OSK</td>
<td>$m^2 \log(\lambda + 1)$</td>
<td>$m$</td>
<td>$m$</td>
<td>$2N \log(s \sqrt{N})$</td>
</tr>
<tr>
<td>PSS</td>
<td>$2m \log(\lambda + 1)$</td>
<td>$m$</td>
<td>$m$</td>
<td>$4N \log(12\sigma) + 2N (\log \lambda + 1)$</td>
</tr>
<tr>
<td>PSK</td>
<td>$m^2 \log(\lambda + 1)$</td>
<td>$7m$</td>
<td>$m$</td>
<td>$2N \log(s \sqrt{N})$</td>
</tr>
</tbody>
</table>

### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

### Acknowledgments

This research is supported by the National Natural Science Foundation of China (no. U1636213).

### References


