Cryptanalysis of Lattice-Based Blind Signature and Blind Ring Signature Schemes

KYUNG-AH SHIM*, (Member, IEEE), AND YOUNGJOO AN
Division of Industrial Mathematics, National Institute for Mathematical Sciences, Daejeon, Yuseong-gu 34047, South Korea
Corresponding author: Kyung-Ah Shim (kashim@nims.re.kr)

This work was supported by the National Institute for Mathematical Sciences funded by the Ministry of Science and ICT of Korea under Grant B21720000.

ABSTRACT A blind signature enables a user to obtain signatures on any message from an authority who cannot acquire any information on the message being signed. A blind ring signature scheme is designed as a ring signature scheme with the blindness property. The scheme allows any member of a group anonymously sign a message on behalf of the group. Also, the user with the message can blind it before transmitting to the group. At Asiacrypt 2010, Rückert constructed the first blind signature scheme using ideal lattices. Recently, Zhang, Jiang and Zheng, and Alkadri, Bansarkhani and Buchmann proposed two improved blind signature schemes based on the SIS problem and the Ring SIS problem in 2018 and 2020, respectively. At WISA 2019, motivated by these blind signature schemes, Le, Duong and Susilo constructed the first lattice-based blind ring signature scheme provably secure under the hardness assumption of the SIS problem in random oracle model. In this paper, we show that Rückert’s scheme, Alkadri-Bansarkhani-Buchmann scheme and Zhang-Jiang-Zheng scheme, and Le-Duong-Susilo scheme do not achieve blindness, i.e. the signer can link a valid message-signature pair after interacting with various users. We show that the cause of vulnerabilities of the blind schemes is that the blinding factors to hide real messages being signed are exposed by specific algebraic relations in the underlying rings. To hide the blinding factors, we use homomorphic encryption schemes. Finally, we propose a generic construction from a semantically secure homomorphic encryption scheme and a one-more unforgeable blind signature scheme that does not achieve blindness to a new blind signature scheme that achieves blindness as well as one-more unforgeability.

INDEX TERMS Anonymity, blind signature, blind ring signature, blindness, homomorphic encryption, lattice-based, unlinkability, SIS problem.

I. INTRODUCTION

Since its invention in the late 1970s [1], public-key cryptography is fundamental building block for secure communications in cyber security. E-commerce, online banking, cloud computing and mobile communication depend on the security of the underlying cryptographic algorithms. Basic security requirements, confidentiality, user authentication, data integrity and non-repudiation are assured by using appropriate public-key and symmetric-key cryptgraphic algorithms. In particular, public-key signature schemes provide non-repudiation, integrity of transmitted messages and authentication. In real-world scenarios, a number of applications need special security requirements other than the basic security requirements. A typical example of the special security requirements is anonymity suitable for electronic voting systems and e-cash system. To meet these demands, the concepts of blind signatures, ring signatures and blind ring signatures have been proposed.

Chaum [2] proposed a notion of blind signature scheme which allows a user to get signatures from an authority on any message, in such a way that the authority learns nothing on the message being signed. Blind signature schemes allow users to generate signatures on messages while interacting with a signer such that the signer obtains no information about the messages being signed (this property is called blindness or unlinkability). The blind signature schemes are suitable for electronic voting systems, electronic auctions and e-cash systems such as Bitcoin.

Rivest et al. [3] proposed a ring signature scheme which provides anonymity. In the ring signature scheme, a signer in a ring consisting of members to sign a message on behalf
of the ring and a verifier can confirm that the signature is given by a public key in the ring, but anyone cannot reveal the real signer’s identity, thus guarantees anonymity of the signer. Ring signatures are useful for whistle blowing [3] and anonymous membership authentication for ad hoc groups [4].

One wants to make a single e-bank system more scalable by supporting many banks and adding some other properties like anonymity of the signing banks and unlinkability of two different signatures in some real-life applications. Various real-life scenarios such as e-cash systems and multi authority e-voting require a combination of blind signatures and ring signatures.

With the threat of a quantum computer capable of running Shor’s quantum algorithm [5] to break currently used public-key cryptographic algorithms, cryptography and security communities have inspired great interest in post-quantum cryptography, where post-quantum cryptography means cryptographic primitives believed to protect both classical and quantum attacks. Post-quantum cryptography have considered as post-quantum replacements, where lattice-based cryptographic algorithms is one of the most promising candidates due to their security under worst-case hardness assumptions. At Asiacrypt 2010, Rückert [6] constructed the first blind signature scheme using ideal lattices. For that reason, a number of lattice-based blind signature schemes and blind signature schemes have been proposed. Recently, Zhang et al. [7] and Alkadri et al.’s scheme [8] proposed improved blind signature schemes based on the SIS problem and the Ring SIS problem in 2018 and 2020, respectively. The first lattice-based ring signature scheme using Lyubashevsky’s signature scheme [9] over ideal lattices was proposed by Aguilar-Melchor et al. [10]. In 2018, Wang et al. [11] proposed a new ring signature scheme using the improved scheme of Lyubashevsky [12]. At WISA 2019, motivated by these signature schemes [6], [7], [11], Le et al. [13] constructed the lattice-based first blind ring signature scheme proven secure under the hardness of the SIS problem in random oracle model. In this paper, we show that Rückert’s scheme [6], Alkadri et al.’s scheme [8] and Zhang et al.’s scheme [7] do not achieve blindness. More precisely, we show that a signer can distinguish the views generated by distinct messages, thus the signer can link a valid message-signature pair after interacting with various users. We also show that Le et al.’s blind ring signature scheme does not achieve blindness.

Organization of this paper is as follows. We describe Rückert’s scheme [6], Alkadri et al.’s scheme [8] and Zhang et al.’s scheme [7] and Le et al.’s scheme [13] in Section II. We show that these four schemes do not achieve blindness and discuss some improvements in Section III. Section IV concludes this paper.

II. PRELIMINARIES

We introduce the definitions of blind signature and blind ring signature schemes and describe three blind signature schemes in [7] and a blind ring signature scheme in [13].

A. BLIND SIGNATURE AND BLIND RING SIGNATURE SCHEMES

1) BLIND SIGNATURE SCHEME

A blind signature scheme is defined by three polynomial-time algorithms (KeyGen, Signature Issuing Protocol, Verify).

- **KeyGen**($\lambda$). For a security parameter $\lambda$, output a secret signing key $sk$ and a public key $pk$.

- **Blind Signature Protocol.** This interactive protocol between a user $U$ and a signer $S$ runs as:
  - **Blind:** A signer and a user send and receive ‘commitment’ and ‘challenge’, respectively.
  - **Sign:** After receiving the challenge, the signer with the secret key $sk$ generates a blind signature $\sigma^*$ on a blind message $m^*$ and transmits it to the user.
  - **Unblind:** For a blind signature $\sigma^*$ of a message $m$ using the secret key $sk$, output a unblinded signature $\sigma$.

- **Verify.** This algorithm Verify($pk, \sigma, m$) returns 1 if $\sigma$ is a valid signature of $m$ for $pk$ and otherwise 0.

2) SECURITY NOTIONS OF BLIND SIGNATURE SCHEME

Security notion of the blind signature scheme captures two properties: blindness and one-more unforgeability.

- **Blindness.** It means that a signer in the blind signing protocol cannot learn any information on the messages the user obtained signatures on, i.e. it means that the signer is unable to link a valid message-signature pair after interacting with various users.

- **One-more Unforgeability.** It is a special type of unforgeability, i.e. the user that has been involved in $l$ runs of the blind signing protocol cannot get more than $l$ signatures.

3) BLIND RING SIGNATURE SCHEME

A blind ring signature scheme is defined by four polynomial-time algorithms, (Setup, KeyGen, Sign, Verify).

- **Setup**($\lambda^*$. For a security parameter $\lambda$, output public parameters $P$.

- **KeyGen**($P$). For the public parameters $P$, return a secret key $sk$ and a public key $pk$ associated to a signer of the ring $R = \{S_1, \ldots, S_l\}$ with PK as a set of public keys in $R$.

- **Sign**($sk_j, P, \mu, PK$). This interactive protocol between a ring of signers $R = \{S_1, \ldots, S_l\}$ and a user $U$ runs as follows:
  - For a message $\mu$, $U$ generates a blind message $\mu^*$ and sends $\mu^*$ to $R$. The ring $R$ selects a member, $S_j$, to interact with the user as the real signer, where $S_j$ possesses a secret key $sk_j$.
  - The signer gets the blinded signature $\Sigma^*$ on $\mu^*$ and returns its view, $\forall$.
  - By un-blinding $\Sigma^*$, $U$ returns $\Sigma$ as a final signature on $\mu$. Note that $U$ may obtain a failure symbol $\bot$.

- **Verify**($\Sigma, P, \mu, PK$). For a signature $\Sigma$ on a message $\mu$, the set of public keys PK and the public parameters $P$, return 1 if $\Sigma$ is valid or 0 otherwise.
4) SECURITY NOTIONS OF BLIND RING SIGNATURE SCHEME

Security notion of the blind ring signature schemes capture three properties: anonymity, blindness and one-more unforgeability. 

- **Anonymity.** This guarantees that, in the blind ring signature protocol, a user cannot know which member of the ring was the real signer.

The other properties are the same as those of the blind signature schemes.

**B. NOTATIONS**

- We denote a column vector as small bold letter and a matrix as bold capital letter.
- Given any positive integer \( q \), we denote \( \mathbb{Z}_q \) as the set of integers in the range \([-q/2, q/2)\cap \mathbb{Z} \).
- Euclidean norm of a vector \( \psi \) with entries \( v_i \) is defined as \( \| \psi \| = (\sum_i |v_i|^2)^{1/2} \) and its \( \ell_\infty \) norm \( \| \psi \|_\infty = \max |v_i| \).
- Notation \( \alpha \rightleftarrows A \) means that \( A \) is sampled uniformly at random from a set \( A \).
- Discrete Gaussian distribution over \( \mathbb{Z}^m \) centered at some \( \psi \in \mathbb{Z}^m \) with standard deviation \( \sigma \) is defined as

\[
\mathcal{D}_{\psi, \sigma}^m(x) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{-\frac{(x-\psi)^2}{2\sigma^2}}
\]

where \( \rho_{\psi, \sigma}^m(x) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{-\frac{(x-\psi)^2}{2\sigma^2}} \) and \( \rho_{\psi, \sigma}^m(\mathbb{Z}^m) = \sum_{x \in \mathbb{Z}^m} \rho_{\psi, \sigma}^m(x) \).

**C. DESCRIPTIONS OF THREE BLIND SIGNATURE SCHEMES**

Here, we describe three blind signature scheme using lattices, Rückert’s scheme [6], Alkadri et al.’s scheme [8] and Zhang et al.’s scheme [7].

1) Rückert’s BLIND SIGNATURE SCHEME

To describe Rückert’s scheme [6], we define the followings. Let \( n \) be a power of 2 and \( d_e \) be a positive integer constant less than \( q/(4m) \), a constant \( c_m \) satisfies \( 1/\log(2d_e) = c_m \log d_e \) + 1, a prime \( q \geq 4mn\sqrt{m} \log(n)d_e \). All sets are subsets of the ring \( \mathbb{R} = \mathbb{Z}_q[X]/(X^n + 1) \) and are defined via a norm bound:

\[
\begin{align*}
D_s &= \{ f \in \mathbb{R} : \| f \|_{\infty} \leq d_s \}, \\
D_e &= \{ f \in \mathbb{R} : \| f \|_{\infty} \leq 1 =: d_e \}, \\
D_A &= \{ f \in \mathbb{R} : \| f \|_{\infty} \leq \psi nd_e \} =: d_A,
\end{align*}
\]

where \( \psi, \phi \) are positive integer constant greater than 1.

- Choose a secret key \( \tilde{s} \rightleftarrows D_s^m \) and a compression function \( h \rightleftarrows H(\mathbb{R}, m) \).
- Choose a commitment function \( \text{com} \rightleftarrows C(1^n) \) and a function \( H \rightleftarrows H(1^n) \) mapping \( [0, 1]^* \to D_e \subset D \).
- Calculate a public key \( S \rightleftarrows h(\tilde{s}) \) and output \( (\tilde{s}, S) \) as a secret/public key pair.

**Signature Protocol.** This interactive protocol between a user \( U \) and a signer \( S \) runs as:

- For a message \( M \in [0, 1]^* \), \( S \) selects \( \tilde{y} \rightleftarrows D_s^m \) and sends a commitment \( Y = h(\tilde{y}) \) to \( U \).
- After receiving \( Y, U \) selects \( r \rightleftarrows [0, 1]^n, \alpha \rightleftarrows D_A, \beta \rightleftarrows D_A^m \), and computes \( e = H(Y - \alpha S - h(\beta)) \) with the commitment \( C = \text{com}(M; r) \in [0, 1]^n \). \( U \) computes \( e^* = e - \alpha \) and sends \( e^* \) to the signer if \( e^* \) is in \( D_e \). This process restarts with a fresh \( \alpha \) if \( e^* \notin D_e \).
- After receiving \( e^*, S \) computes \( \tilde{z}^* \in G_m \) and send it to \( U \) if \( \tilde{z}^* \notin G_m \). Whenever \( S \) triggers a restart, \( S \) computes a new \( \tilde{z}^* \) for \( e^* \) if \( e^* \notin D_e \).
- On receiving \( \tilde{z}^*, U \) computes \( \tilde{z} = \tilde{z}^* - \beta \). If \( \tilde{z} \in G_m \), the user outputs a signature \( (r, \tilde{z}, e) \).

Otherwise, \( U \) computes \( \text{result} = (C, \alpha, \beta, e) \) and sends to \( S \). After receiving \( \text{result} \), \( S \) restarts the signing protocol after checking the following conditions:

\[
e^* + \alpha = e = H(Y - \alpha S - h(\beta)), \\
H(h(\tilde{z}^* - \beta) - \hat{S}, C) = e, \quad \hat{z}^* - \beta \notin G_m^m.
\]

**Verify(S, (r, \tilde{z}, e), M).** This verification algorithm outputs \( 1 \) if \( \tilde{z} \in G_m \) and \( H(h(\tilde{z}) - \hat{S}, \text{com}(M; r)) = e \).

2) ZHANG et al.’s BLIND SIGNATURE SCHEME

**Key Generation.** For a security parameter \( \lambda \), this algorithm generates a secret/public key as:

- Choose \( A \rightleftarrows \mathbb{Z}_{2^\lambda}^{m \times k} \) and \( S \rightleftarrows \{-d, \ldots , 0, \ldots , d\}^{m \times k} \), where \( d \) is a small positive integer.
- Compute \( T \rightleftarrows AS \), where \( T \) is an \( n \times k \) matrix.
- Output \((A, T)\) as a public key and \( S \) as a secret key.

**Signature Protocol.** This interactive protocol between a signer \( S \) and a user \( U \) runs as:

- To sign a message \( \mu \), \( S \) selects a random vector \( r \rightleftarrows D_{m \times 2}^m \) and sends a commitment \( x \rightleftarrows \text{Ar} \).
- After receiving \( x, U \) picks blind factors \( a \rightleftarrows D_{m \times 2}^m \), \( b \rightleftarrows D_{m \times 2}^m \) and computes \( x + Aa + Tb \) and \( C = \text{com}(\mu, t) \). Then \( U \) computes \( c = H(x + Aa + Tb, C) \) transmits \( e = c + b \) to \( S \) with some probability, where \( H : [0, 1]^* \to [\{v \in [-1,0,1]^k, \|v\|_1 \leq k\}] \) is a hash function.
- After getting \( e, S \) calculates \( y = r + Se \) and transmits it to \( U \).
- On the receipt of \( y, U \) outputs \( z = y + a \) with the probability \( D_{m \times 2}^m(x)/M \_{D_{m \times 2}^m}^m(z) \). If \( z \) is not in the
rejection area \( J, \mathcal{U} \) outputs a final signature

\[
(\mu, (z, c, t)).
\]

Otherwise, \( \mathcal{U} \) computes result = (a, b, com(\mu, t)) and sends to the signer \( S \). \( S \) restarts the signing protocol after checking the following conditions: parse

\[
\text{result} = (a, b, c),
\]

\[
e - b = c = H(x + Aa + Tb, \text{com}(\mu, t)),
\]

\[
c = H(Aa + Ay - tc, \text{com}(\mu, t)),
\]

\[
y + a \in J.
\]

- **Verification.** Verify(A, T, \( \mu, z, c, t \)) = 1 iff

\[
\|z\| \leq \eta \sqrt{m_3}, c = H(Az - Tc, \text{com}(\mu, t)),
\]

where \( \eta > 1 \), so that the probability of verification algorithm is about \( 1 - 2^{-k} \).

3) **BLAZE**

To describe BLAZE, we define the following functions:

- **Expand** is a random function that expands a uniform random seed in \( \{0, 1\}^k \) to any desired length for the security parameter \( \lambda \).

- **Com**: \( \{0, 1\}^k \times \{0, 1\}^k \to \{0, 1\}^k \) is a computational binding and statistically hiding commitment function.

- **H**: \( \{0, 1\}^k \to T_k^n \) is a cryptographic hash function.

- **Compress** and **Decompress** are functions to (de)compress Gaussian elements.

BLAZE runs as follows:

- **Key Generation.** For a security parameter \( \lambda \), this algorithm generates a secret/public key pair as:

  - Select a random seed \( \lambda \) and expand it to a polynomial \( \hat{a} \in R_q \) using the Expand function.

  - A secret key consists of two polynomials \( sk = (\hat{s}_2, \hat{s}_2) \) chosen from \( D_{\mathbb{Z}, \sigma}^n \) and a public key is given by

\[
\text{pk} = (\text{seed}, \hat{b} = \hat{a}s_1 + \hat{s}_2 \mod q).
\]

- **Signing.** This interactive protocol between a signer \( S \) and a user \( \mathcal{U} \) runs as:

  - Given a message \( \mu \), seed and \( sk \), \( S \) samples \( 2\kappa \) masking terms \( \hat{y}_1, \hat{y}_2 \) from \( D_{\mathbb{Z}, \sigma}^n \) for \( j \in [k] \) and transmits \( \hat{y}_j = \hat{a}y_{j,1} + \hat{y}_{j,2} \mod q \) to \( \mathcal{U} \).

  - On the receipt of \( \hat{y}_1, \ldots, \hat{y}_k \), \( \mathcal{U} \) computes commitments

\[
\tau = \text{Com}(\mu; r), \quad \tau’ = \text{Com}(\rho; r’)
\]

for uniformly random \( r, r’, \rho \) from \( \{0, 1\}^k \), expands \( \text{seed} \) to the polynomial \( \hat{a} \) and chooses uniformly random elements \( \hat{p}_1, \ldots, \hat{p}_k \in \hat{T} \). Furthermore, \( \mathcal{U} \) samples two polynomials \( \hat{e}_1, \hat{e}_2 \) from \( D_{\mathbb{Z}, \sigma}^n \) using a randomness \( \rho \) for the reduction of communication complexity: a proof of failure from \( \mathcal{U} \) contains only \( \rho \) rather than \( \hat{e}_1, \hat{e}_2 \). Then \( \mathcal{U} \) generates

\[
\hat{c} = H(\hat{a}\hat{e}_1 + \hat{e}_2 + \sum_{l=1}^{\kappa} \hat{p}_l \hat{y}_l \mod q, \tau’, \tau) \in T_k^n
\]

and \( \mathcal{U} \) splits \( \hat{c} \) into partitions \( \hat{c}_1, \ldots, \hat{c}_k \in \hat{T} \) so that \( \hat{c} = \sum_{a=1}^{k} \hat{c}_j \) and the j-th partition \( \hat{c}_j \) includes the j-th non-zero entry of \( \hat{c} \) at the same position. Then, \( \mathcal{U} \) masks each partition \( \hat{c}_j \) by calculating

\[
\hat{c}_j^\ast = \hat{p}_j^{-1} \cdot \hat{c}_j
\]

for all \( j \in [k] \) and transmits \( \hat{c}_1^\ast, \ldots, \hat{c}_k^\ast \) to \( S \).

- Utilizing the partitions \( \hat{c}_j^\ast \), \( S \) calculates

\[
\hat{z}_{j,1}^\ast = \hat{y}_{j,1}^\ast \hat{e}_j^\ast, \quad \hat{z}_{j,2}^\ast = \hat{y}_{j,2}^\ast + \hat{s}_2 \hat{c}_j^\ast.
\]

Next, \( S \) uses the rejection sampling on \( \hat{z}_{j,1}^\ast, \hat{z}_{j,2}^\ast \) to prevent the leakage of information on \( sk \). If \( \text{RejSamp} \) outputs 1, then \( S \) transmits \( \hat{z}_{j,1}^\ast, \hat{z}_{j,2}^\ast \) to the user, otherwise \( S \) restarts the protocol.

- On the receipt of \( \hat{z}_{j,1}^\ast, \hat{z}_{j,2}^\ast \) for \( j \in [k] \), \( \mathcal{U} \) calculates

\[
\hat{v}_1 = \sum_{i=1}^{\kappa} \hat{p}_i \hat{z}_{j,1}^\ast, \quad \hat{v}_2 = \sum_{i=1}^{\kappa} \hat{p}_i \hat{z}_{j,2}^\ast
\]

and checks that

\[
\| (\hat{v}_1, \hat{v}_2) \| \leq \eta s^* \sqrt{2k\kappa}.
\]

Then \( \mathcal{U} \) computes

\[
\hat{z}_1 = \hat{e}_1 + \hat{v}_1, \quad \hat{z}_2 = \hat{e}_2 + \hat{v}_2.
\]

If \( \text{RejSamp}(\hat{z}_1, \hat{z}_2; \rho) = 1 \), \( \mathcal{U} \) computes \( (\hat{z}_1, \hat{z}_2) \) from **Decompress** and outputs a signature

\[
(\mu, (\tau’, r, \hat{z}_1, \hat{z}_2, \hat{c})).
\]

Otherwise, \( \mathcal{U} \) computes

\[
\text{result} = (\tau, \rho, r’, \hat{p}_1, \ldots, \hat{p}_k, \hat{c})
\]

and sends to \( S \).

- The signer \( S \) restarts the signing protocol after checking the following:

\[
T’ = \text{Com}(\rho’; r’), \quad \hat{e}_1, \hat{e}_2 \leftarrow D_{\mathbb{Z}, \sigma}^n(\rho’).
\]

\[
\hat{z}_1 = \hat{e}_1 + \sum_{i=1}^{\kappa} \hat{p}_i \hat{z}_{j,1}^\ast, \quad \hat{z}_2 = \hat{e}_2 + \sum_{i=1}^{\kappa} \hat{p}_i \hat{z}_{j,2}^\ast,
\]

\[
\sum_{i=1}^{\kappa} \hat{p}_i \hat{c}_j^\ast = \hat{c} = H(\hat{a}\hat{e}_1 + \hat{e}_2 + \sum_{i=1}^{\kappa} \hat{p}_i \hat{y}_l \mod q, T’, \tau),
\]

\[
\hat{c} = H(\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c} \mod q, T’, \tau),
\]

\[
\text{RejSamp}(\hat{z}_1, \hat{z}_2; \rho’) = 0.
\]

- **Verification.** For (seed, \( \hat{b}, \mu, (\tau’, r, \hat{z}_1, \hat{z}_2, \hat{c}) \)) the verifier utilizes Expand to calculate \( \hat{a} \) out of seed, decompresses \( (\hat{z}_1, \hat{z}_2) \) using **Decompress**. It accepts iff \( \| (\hat{z}_1, \hat{z}_2) \| < B \), where \( B \) is the predefined bound and

\[
H(\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c} \mod q, T’, \text{com}(\mu; r)) = \hat{c}.
\]
D. DESCRIPTIONS OF LE et al.’S BLIND RING SIGNATURE SCHEME

Here, we describe Le et al.’s blind ring signature scheme based on the SIS problem [13].

- **BRS.Setup**($1^n$). For a security parameter $n$, this algorithm returns parameters $\mathcal{P}$.  
- **BRS.KeyGen**. For a ring $R = \{S_1, \ldots, S_l\}$ of $l$ members, this algorithm generates a public key and a secret key of each member as:
  - By running $l$ times **BRS.KeyGen** which uses the preimage sample functions, for given a matrix $T \leftarrow \mathbb{Z}_q^{m \times k}$, generate a secret key $S_i \in \mathbb{Z}_q^{m \times k}$ for each signer $S_i$ in $R$ such that $T = A_i S_i$, where $A_i \leftarrow \mathbb{Z}_q^{m \times m}$ and then $(T, A_i)$ is a public key. The secret key $S_i$ follows a discrete Gaussian distribution $\mathcal{D}_{\sigma_i}^{m \times k}$.
- **BRS.Sign**. This interactive protocol between a ring $R = \{S_1, \ldots, S_l\}$ and a user $\mathcal{U}$ runs as follows, where $\mathcal{U}$ knows that, to interact with $\mathcal{U}$, the ring secretly selects some signer $S_j$ in $R$.
  - **Phase 1**: According to the distribution $\mathcal{D}_{\sigma_j}^m$, a signer samples randomly a list $\{s_i\}_{i \in [l]}$ and transmits a commitment $x = \sum_{i \in [l]} A_i s_i$ to $\mathcal{U}$.
  - **Phase 2**: The user $\mathcal{U}$ selects a vector $t \leftarrow \mathcal{D}_{\sigma_j}^m$ and computes a commitment $C := \text{com}(\mu, t) \in \{0, 1\}^n$. The user $\mathcal{U}$ also chooses blind factors $a_i \leftarrow \mathcal{D}_{\sigma_i}^k$ for all $i \in [l]$ and $b \leftarrow \mathcal{D}_{\sigma_j}^k$ and calculates $u = x + \sum_{i \in [l]} A_i a_i + Tb, \ c = H(u, C)$.
  - **Phase 3**: In this signing phase, the signer $S_j$ using the secret key $S_j$ computes a partial signature $y_j = s_j + S_j e$ on $\mu$ and set $y_i = s_i$ for $i \in [l] \setminus \{j\}$. To prevent the leakage of information on the secret key $S_j$, $S_j$ utilizes the rejection sampling to obtain a blinded challenge $e$. At last, $\mathcal{U}$ transmits $e$ to $R$.
  - **Phase 4**: The user $\mathcal{U}$ calculates $z_i = y_i + a_i$ for all $i \in [l]$. To guarantee that $z_i$ is independent of $y_i$ for blindness, the rejection sampling is utilized. If $\|z_i\| \leq \eta \sqrt{n_0}$ for all $i \in [l]$ then $\mathcal{U}$ returns a final signature $(\mu, \Sigma = ((z_i)_{i \in [l]}, c, t))$.
  - **Phase 5**: If the signer obtains the validity confirmation from $\mathcal{U}$, he returns the view $V = (x, e, (s_i, y_i)_{i \in [l]})$.

If the signer obtains result $\neq$ accept, he restarts the protocol after checking the conditions.

III. CRYPTANALYSIS OF BLIND SIGNATURE AND BLIND RING SIGNATURE SCHEMES

Now, we show the three blind signature schemes in [6]–[8] and the blind ring signature scheme in [17] do not achieve blindness: a signer can distinguish the views generated by distinct messages.

A. CRYPTANALYSIS OF THREE LATTICE-BASED BLIND SIGNATURE SCHEMES

In the blind signature schemes, blindness is a fundamental feature: the signer can only see views which are independent of the message being signed, in other words, the views generated by two distinct messages are indistinguishable. Rückert [6], Alkadri et al. [8] and Zhang et al. [7] proved that their blind signature schemes were statistically blind. They claimed that the signer cannot link a valid message-signature pair after interacting with various users (unlinkability). We show that the schemes do not achieve blindness (unlinkability) unlike their claims: a signer can distinguish the views generated by distinct messages. A main idea of our attacks is that the exposure of blinding factors in a given view and a blind signature makes it possible to determine linkability by checking the validity of the related verification equation.

Adversarial Model of Blind Signature Schemes for Blindness. An adversarial model of the blind signature schemes for blindness is defined by a game between an adversary $A$ and a challenger $C$. $A$’s advantage $Adv_{BS, A}$ is determined by the probability of success as:

- **Setup**. After running the **KeyGen** algorithm, $C$ gives $A$ the resulting system parameters.
- **Sign-View Queries**. When $A$ wants to get view-signature pairs, $C$ outputs the view-signature pairs from the **Blind Signature Protocol** oracle.
- **Output**. At last, $A$ returns a linked view-signature pair $(\overline{V}, \overline{\sigma})$ on a message $\overline{m}$ and wins the game if $Vfy(pk, m, \sigma) = 1$.

We present attacks on the four blind signature schemes to break blindness. In the blind signature schemes, there are three entities, a signer, a user and a verifier, where a signer who signs a blinded message generate a blinded signature on the blinded message, a user generates a unblinded signature on the unblinded real message from the signer’s a blinded signature and a verifier checks the validity of the user’s unblinded signature. In the above adversarial model of the blind signature schemes, an adversary can access a number of view-signature pairs as much as he wants since the adversary can access the blind signature signing oracle. The adversary can access view-signature pairs validly generated by each
entity according to the protocol specifications. We show that via the attacks, the adversary including the signer can link a valid message-signature pair after interacting with various users.

1) LINKABILITY OF RÜCKERT’S BLIND SIGNATURE SCHEME
- Suppose that a pair of a view, \(\mathcal{V} = (\mathcal{Y}, e^*, \hat{z}^*)\), and a signature on \(M, \sigma = (M, r, \hat{z}, e)\) is given. We want to determine its linkability.
- If it satisfies the following equation
  \[
  H(\mathcal{Y}, S(e - e^*) - h(\hat{z}^* - \hat{z}), C) = e \tag{1}
  \]
  then the pair is linked.
- The signature comes from the view if and only if it satisfies
  \[
  H(\mathcal{Y}, S(e - e^*) - h(\hat{z}^* - \hat{z}), C)
  = H(\mathcal{Y}, S\alpha - h(\beta), C) = e,
  \]
  since
  \[
  e - e^* = \alpha, \hat{z}^* - \hat{z} = \beta, \text{ and } C = com(M, r).
  \]
  Thus, if the equation (1) holds, then the pair is linked.
- In the scheme, the user utilizes \(\alpha\) and \(\beta\) to generate the blinded message and its final signature. Anyone can compute the blending factors \(\alpha\) and \(\beta\) from the given view and the signature that makes it possible to determine linkability by checking the validity of the related verification equation.

2) LINKABILITY OF BLAZE
- Given a view \(\mathcal{V} = (\hat{y}, \hat{e}, z^*, \tau, \hat{r}, \rho', r, \hat{p}_1, \cdots, \hat{p}_k, \hat{c})\) and a signature on \(\mu, \sigma = (\mu, (\tau', \hat{r}, \hat{z}_1, \hat{z}_2, \hat{c}))\), if it satisfies the following equation
  \[
  H(\hat{a}\hat{z}_1 - \sum_{j=1}^{k} \hat{p}_j\hat{z}_{j,1}) + \hat{z}_2 - \sum_{j=1}^{k} \hat{p}_j\hat{z}_{j,2}
  + \sum_{j=1}^{K} \hat{p}_j\hat{y}_j (\mod q), \tau', \tau) = \hat{c} \tag{2}
  \]
  then the pair \((\mathcal{V}, \sigma)\) is linked.
- The signature \(\sigma\) comes from the view if and only if it satisfies
  \[
  H(\hat{a}\hat{z}_1 - \sum_{j=1}^{k} \hat{p}_j\hat{z}_{j,1}) + \hat{z}_2 - \sum_{j=1}^{k} \hat{p}_j\hat{z}_{j,2}
  + \sum_{j=1}^{K} \hat{p}_j\hat{y}_j (\mod q), \tau', \tau)
  = H(\hat{a}\hat{c}_1 + \hat{c}_2 + \sum_{j=1}^{K} \hat{p}_j\hat{y}_j (\mod q), \tau', \tau) = \hat{c}
  \]
  since the values
  \[
  <\hat{y} = (\hat{y}_1, \cdots, \hat{y}_k), \hat{z}^* = (\hat{z}_{1.1}, \cdots, \hat{z}_{2,k}), \hat{p}_1, \cdots, \hat{p}_k, \tau >
  \]
  are given in the view, \(<\tau', \hat{c}>\) are given in the signature, and
  \[
  \hat{z}_1 - \sum_{j=1}^{k} \hat{p}_j\hat{z}_{j,1} = \hat{c}_1, \hat{z}_2 - \sum_{j=1}^{k} \hat{p}_j\hat{z}_{j,2} = \hat{c}_2.
  \]
  Thus, if the equation (2) holds, then the pair is linked.

3) LINKABILITY OF ZHANG ET AL.’S BLIND SIGNATURE SCHEME
- For a pair of a view, \(\mathcal{V} = (x, (e, y))\), and a signature on \(\mu, \sigma = (\mu, (z, c, t))\), if it satisfies the following equation
  \[
  H(x + A(z - y) + T(e - c), com(\mu, t)) = c \tag{3}
  \]
  then the pair \((\mathcal{V}, \sigma)\) is linked.
- The signature \(\sigma\) comes from the view if and only if it satisfies
  \[
  H(x + A(z - y) + T(e - c), com(\mu, t))
  = H(x + Aa + Tb, com(\mu, t)) = c
  \]
  since \(e - c = a\) and \(z - y = b\). Thus, if the equation (3) holds then the pair is linked.

B. CRYPATANALYSIS OF LE ET AL.’S BLIND RING SIGNATURE SCHEME
Here, we show that Le et al.’s blind ring signature scheme does not achieve blindness. More precisely, given a view \(\mathcal{V}\) and a signature \(\sigma\) generated by a signer in the scheme, the signer or anyone can determine whether \(\sigma\) comes from \(\mathcal{V}\) or not. Thus, the signer can link a valid message-signature pair after interacting with various users.
- For a view, \(\mathcal{V} = (x, e, \Sigma^* = \{y_i\}_{i \in [l]}\), and a signature on \(\mu, \sigma = ((\xi_i)_{i \in [l]}, c, t))\), if it satisfies the following equation
  \[
  H(\sum_{i \in [l]} A_i(z_i - y_i) + T(e - c) + x, \ com(\mu, t), PK) = c \tag{4}
  \]
  then the pair \((\mathcal{V}, \sigma)\) is linked.
- The signature \(\sigma\) comes from the view if and only if it satisfies
  \[
  H(\sum_{i \in [l]} A_i(z_i - y_i) + T(e - c) + x, \ com(\mu, t), PK)
  = H(\sum_{i \in [l]} A_i a_i + Tb + x, \ com(\mu, t), PK)
  = H(w + Tb + x, \ com(\mu, t), PK)
  = H(u, \ com(\mu, t), PK) = c,
  \]
  since \((z_i - y_i) = a_i\) and \(e - c = b\). Thus, if the equation (4) holds then the pair is linked.

C. DISCUSSIONS ON SOME IMPROVEMENTS
In this section, we discuss why the blind signature schemes do not achieve blindness, and two countermeasures for achieving blindness. We then propose a generic construction for blindness.
1) DISCUSSIONS

In the blind signature schemes [6]–[8], [13], we show that the cause of vulnerabilities of the blind schemes against our attacks is that the blinding factors to hide real messages being signed are exposed by specific algebraic relations in the underlying rings.

- The blinding factors ($\alpha$, $\beta$) in Rückert’s scheme [6] and ($a$, $b$) in Zhang et al.’s scheme [7], can be computed from the views and the signatures and then the linkability can be determined by the known algebraic relations. If one can prevent this exposure of the blinding factors then blindness can be achieved. The corresponding blind factors and the used algebraic relations to recover the blind factors form the view-signature pairs in our attacks on the four blind signature schemes are summarized in TABLE 1.

To prevent our attacks, it needs a method to hide the blinding factors. There are two ways to hide the blinding factors as:

- If one can find appropriate algebraic relations to make the blinding factors hidden then blindness can be achieved.
  - In Camenisch et al.’s two blind signature schemes [14], one was based on the modified Digital Signature Standard (DSA) [15] and the other was based on Nyberg-Rueppel signature scheme [16]. They successfully hid the blinding factors $a$ and $b$ used in the blinded messages and their related signatures using some algebraic relations in the underlying groups.

- If one encrypts the hiding factors using a public-key encryption scheme and their operations are preserved then it is possible to construct blind signature schemes secure against our attacks. Such a public-key encryption scheme is a homomorphic encryption.
  - At Asia CCS 2019 [17], Yi and Lam proposed a new blind signature scheme based on the international standard signature scheme, ECDSA. Their scheme used a variant of the Paillier cryptosystem [18] and its homomorphic properties to achieve the blind property. They showed that if the modified Paillier cryptosystem is semantically secure, then their blind signature scheme achieves blindness and one-more unforgeability.

2) GENERIC CONSTRUCTION FOR BLINDNESS

Now, we propose a generic construction from a semantically secure homomorphic encryption scheme and a one-more unforgeable blind signature scheme that does not achieve blindness to a new blind signature scheme that achieves blindness as well as one-more unforgeability. To hide the blinding factors, we use homomorphic encryption schemes: i) the homomorphic encryption scheme can encrypt the related information so that the blinding factors cannot be recovered from the ciphertexts by using the algebraic relations and ii) the homomorphic encryption scheme can allow to a valid unblinded signature from the ciphertext of a blinded signature since the scheme preserves the ring operations in the signatures.

Let $BS := (\text{Setup}, \text{KeyGen}, \text{Signature Issuing Protocol}, \text{Verify})$ be a blind signature scheme that does not achieve blindness and $\mathcal{HE} = (\text{KeyGen}, \text{Enc}, \text{Dec})$ be a homomorphic encryption whose operations are compatible with the operations defined on the underlying ring or group of $BS$. A new blind signature scheme $BS \ast \mathcal{HE}$ is constructed as follows:

- **KeyGen** ($\lambda$). For a security parameter $\lambda$, output a secret/public key pair ($sk$, $pk$) by running the KeyGen algorithm of $BS$ and a secret/public key pair ($SK$, $PK$) by running the KeyGen algorithm of $\mathcal{HE}$.

- **Blind Signature Protocol**. This interactive protocol between a user $U$ and a signer $S$ runs as:
  - Blind: A signer send ‘commitment’ to a user. After receiving the commitment, a user generates a challenge, computes $C' = \text{Enc}(PK, Ch, m)$ and sends $C'$ to the signer.
  - Sign: After receiving the encrypted challenge and message, the signer with the signing secret key $sk$ generates a blind signature $\sigma^*$ on $C'$ and transmits it to the user.
  - Unblind: For a blind signature $\sigma^*$ on the encrypted message, using the secret key $SK$, a user decrypts $\sigma = \text{Dec}(SK, \sigma^*)$ and output a unblinded signature $\sigma$.

- **Verify**. This algorithm Verify($pk$, $\sigma$, $m$) returns 1 if $\sigma$ is a valid signature of $m$ for $pk$ and otherwise 0.

In our generic construction, the blind factors can be encrypted by the homomorphic encryption scheme and valid unblinded signatures can be generated due to the following properties

$$Sign(sk, \text{Enc}(PK, Ch, m)) = \text{Enc}(PK, \text{Sign}(sk, Ch, m)).$$

Thus, if the homomorphic encryption scheme $\mathcal{HE}$ is semantically secure, then the blind signature scheme constructed from the generic method achieves blindness.
Theorem 1: If the underlying homomorphic encryption scheme $HE$ is semantically secure (indistinguishability under chosen-plaintext attacks (IND-CPA)) and the blind signature scheme $BS$ achieves one-more unforgeability then the resulting blind signature scheme $BS \ast HE$ from our generic construction achieves blindness and one-more unforgeability.

Proof: It is known that the ciphertext indistinguishability under chosen-plaintext attacks is equivalent to semantic security in probabilistic encryption schemes [19]. This indistinguishability of the homomorphic encryption scheme $HE$ induces indistinguishability between the ciphertexts of the views generated by distinct messages and random distributions. One-more unforgeability of $BS \ast HE$ is preserved from one-more unforgeability of $BS$ and the homomorphic property of $HE$.

As seen Theorem 1, if one uses an efficient and secure lattice-based homomorphic encryption scheme to hide the blinding factors in the blind signature schemes then their improved schemes from the generic construction can achieve blindness. However, the improved schemes require additional overhead for encryptions and decryptions to generate views. Thus, it remains an open problem to construct secure lattice-based blind signature and lattice-based blind ring signature schemes without using the homomorphic encryption schemes.

IV. CONCLUSION

We have shown that the three lattice-based blind signature schemes [6]–[8] and the blind ring signature scheme [17] do not achieve blindness. Their vulnerabilities against our attacks are due to the exposure of the blinding factors used in the blinded messages and their related signatures. We have proposed a generic construction from a semantically secure homomorphic encryption scheme and a one-more unforgeable blind signature scheme which does not achieve blindness, whose resulting blind signature scheme achieves blindness as well as one-more unforgeability. However, the improved schemes are very inefficient since they require additional overhead for encryptions and decryptions to generate views. Thus, we can say that it remains an open problem to construct secure and efficient lattice-based blind signature and lattice-based blind ring signature schemes based on algebraic relations for hiding the blind factors without using the homomorphic encryption schemes.

REFERENCES

[1] W. Diffie and M. Hellman, “New directions in cryptography,” IEEE Trans. Inf. Theory, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.
[2] D. Chaum, “ Blind signature for untraceable payment,” in Advances in Cryptology-CRYPTO. Santa Barbara, CA, USA: Plenum, Aug. 1982, pp. 199–203.
[3] R. Rivest, A. Shamir, and Y. Tauman, “ How to leak a secret,” in Advances in Cryptology-ASIACRYPT. Gold Coast, QLD, Australia: Springer, Dec. 2001, pp. 552–565.
[4] E. Bresson, J. Stern, and M. Szudy, “ Threshold ring signatures and applications to ad-hoc groups,” in Advances in Cryptology-CRYPTO. Santa Barbara, CA, USA: Springer, Aug. 2002. pp. 465–480.
[5] P. W. Shor, “ Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” SIAM J. Comput., vol. 41, no. 2, pp. 1484–1509, 1997.
[6] M. Rückert, “Lattice-based blind signatures,” in Advances in Cryptology-ASIACRYPT. Singapore: Springer, Dec. 2010, pp. 413–430.
[7] P. Zhang, H. Jiang, Z. Zheng, P. Hu, and Q. Xu, “ A new post-quantum blind signature from lattice assumptions,” IEEE Access, vol. 6, pp. 27251–27258, 2018.
[8] N. A. Alkadiri, R. E. Barsakhani, and J. Buchmann, “ BLAZE: Practical lattice-based blind signatures for privacy-preserving applications,” in Financial Cryptography and Data Security, Kota Kinabalu, Malaysia: Springer, Feb. 2020, pp. 484–502.
[9] V. Lyubashevsky, “ Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures,” in Advances in Cryptology-ASIACRYPT. Tokyo, Japan: Springer, Dec. 2009, pp. 598–616.
[10] C. Aguilar-Melchor, S. Bettaih, X. Boyen, L. Fousse, and P. Gaborit, “Adapting Lyubashevsky’s signature schemes to ring signature setting,” in AFRICACRYPT. Cairo, Egypt: Springer, Jun. 2013, pp. 1–25.
[11] S. Wang, R. Zhao, and Y. Zhang, “ Lattice-based ring signature scheme under the random oracle model,” Int. J. High Perform. Comput. Netw., vol. 11, no. 4, pp. 332–341, 2018.
[12] V. Lyubashevsky, “Lattice signatures without trapdoors,” in Advances in Cryptology-EUROCRYPT. Cambridge, U.K.: Springer, Apr. 2012, pp. 738–755.
[13] H. Q. Le, D. H. Duong, and W. Susilo, “Blind ring signature based on the short integer solution problem,” in WISA (Lecture Notes in Computer Science), vol. 11897. Jeju Island, South Korea: Springer, Aug. 2019, pp. 92–111.
[14] J. Camenisch, J.-M. Piveteau, and M. Stadler, “Blind signatures based on the discrete logarithm problem,” in Advances in Cryptology-EUROCRYPT. Perugia, Italy: May 1994, pp. 428–432.
[15] Nat. Inst. Standards Technol. (NIST), “ Digital signature standard (DSS),” TR. FIPS PUB XX, NISS, US Dept. Commerce, Washington, DC, USA, 1993.
[16] K. Nyberg and R. A. Rueppel, “A new signature scheme based on the DSA giving message recovery,” in Proc. 1st ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, Nov. 1993, pp. 58–61.
[17] Y. Xi and K.-Y. Lam, “A new blind ECDSA scheme for bitcoin transaction anonymity,” in Proc. Asia CCS, Auckland, New Zealand, Jul. 2019, pp. 613–620.
[18] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in Advances in Cryptology-EUROCRYPT. Prague, Czech Republic: Springer, May 1999, pp. 223–238.
[19] S. Goldwasser and S. Micali, “ Probabilistic encryption & how to play mental poker keeping secret all partial information,” in Proc. 14th Annu. ACM Symp. Theory Comput. (STOC), May 1982, pp. 365–377.

KYUNG-AH SHIM (Member, IEEE) received the Ph.D. degree in mathematics from Ewha Womans University, Seoul, South Korea. She is currently a Senior Researcher with the National Institute for Mathematical Sciences. Her research interests include public-key cryptography, post-quantum cryptography, cryptographic protocols, and information security.

YOUNGJOO AN received the B.S., M.S., and Ph.D. degrees in mathematics from Ewha Womans University, Seoul, South Korea, in 2008, 2010, and 2019, respectively. She is currently a Post-doctoral Researcher with the National Institute for Mathematical Sciences, Daejeon, South Korea. Her current research interests include post-quantum cryptography, lattice-based cryptography, and information security.