A Blind Signature from Module Latices

1st Huy Quoc Le
School of Computing and Information Security
University of Wollongong
NSW 2522, Australia
qhl576@uowmail.edu.au

2nd Willy Susilo
Institute of Cybersecurity and Cryptology
School of Computing and Information Security
University of Wollongong
NSW 2522, Australia.
wsusilo@uow.edu.au

3rd Thanh Xuan Khuc
Institute of Cryptography Science and Technology
Government Information Security Committee
Ha Noi, Viet Nam
khucxuanthanh@gmail.com

4th Minh Kim Bui
Ho Chi Minh University of Science,
Vietnam National University
Ho Chi Minh City, Viet Nam
kmath93@gmail.com

5th Dung Hoang Duong
Institute of Cybersecurity and Cryptology
School of Computing and Information Security
University of Wollongong
NSW 2522, Australia.
hduong@uow.edu.au

***Abstract*—Since its birth by Chaum, blind signatures have become one of the fundamental components in the so-called e-cash or e-voting. The scheme ensures a message to be blinded before being signed. Among all lattice-based signatures submitted to NIST post-quantum cryptographic standardization, Dilithium is a very promising candidate. The scheme has some advantages such as simple to implement securely, conservative with parameters and minimal in total size of public key and signature. In this paper, we propose a blind signature scheme based on the framework of Dilithium in order to take advantages of Dilithium. The proposed scheme is blind and one-more unforgeable secure in the random oracle model under the hardness of the module Learning with Errors problem MLWE and the module short integer solution problem MSIS.**

***Index Terms*—Blind Signatures, Module Lattices, standardization**

I. Introduction

A. Background

In the purpose of mimicking the privacy features (e.g. untraceability, unlinkability, no double-spending) of the real coins or cashes, Chaum [Cha83] proposed derivatives of digital signatures, named *blind signatures*. Blind signatures enable a user to make his messages invisible to a signer before the messages are sent to the signer to be signed. This property is useful and necessary in various application scenarios, especially, for example, in electronic system payments (e-cashes) or online e-voting systems, to name a few.

The first blind signature schemes based on RSA and the hardness of the factoring problem were proposed by Chaum [Cha83], [Cha84], [Cha88] aiming to create e-cash systems. He also proposed methods to avoid the double-spending problem. One of those methods is an RSA-based online e-system where the number of spending of a coin is under control by the bank who posses the customer's databases. However, such an e-system was impractical due to the huge of databases and hence the very long runtime of checking if the coin has been spent or not. Chaum then used an RSA-based offline blind signature version. In 1994, Camenisch et al. [CPS95] proposed blind signature schemes based on the discrete logarithm problem one of which is a modification of Digital Signature Algorithm standardized by NIST. The other is a blinded version of the Nyberg-Rueppel digital signature scheme [NR93].

The gradually recent realization of quantum computers warns us about the collapse in the near future of the current cryptographic systems of which security based on the problems still hard against the classical computers but will be easy to solve by quantum ones, as shown by Shor [Sho97]. Among others, lattices promises a solution to the quantum-related problem. In fact, the hard lattice problems, such as the shortest vector problem, are believed to be hard against even the strength of large-scale quantum computers which can break easily cryptographic systems that based on number-theoretic assumption like factoring problem or discrete logarithm problem. The first lattice-based blind signature scheme was proposed by Rückert [Rüc10]. More specifically, [Rüc10] showed a method to construct blind signatures scheme from the framework of Lyubashevsky's identification scheme. The blind signature scheme of Rückert are provably unconditionally blind and one-more unforgeable secure in the random oracle model.

Many blind signature schemes are constructed by modifying an existing digital signature schemes. Thus, the efficacy of the blind signature schemes can be inherited from the digital counterparts. Dilithium [DLL+17], [DKL+19], one of Round 2 candidates for NIST Post-quantum Standardization, is a compact and very efficient digital signature scheme. The construction of Dilithium is based on the rejection sampling, uniform sampling (instead of the expensive Gaussian sampling) and an additional technique to compress the public key. As a result, the total size of public key and signature size of Dilithium is the smallest of any signature scheme based on lattices that only uses uniform sampling. This makes Dilithium become a very promising candidate among others.

B. Contribution

Our main contribution in this paper is to construct a blind signature scheme aiming to base on and take advantages of Dilithium. The difficulty of constructing a blind signature scheme from the framework of Dilithium is from the fact that we need some more materials to blind the message from the signer as well as to un-blind the blinded signature. Indeed, the only hint from the signer as in Dilithium is not sufficient for the correctness of the blind signature scheme. Our measure to the problem is giving one more hint coming from the user. The hint is defined from the materials used for blinding and un-blinding.

II. Preliminaries

A. Notations

Throughout the paper, we follow most of the notations from [DLL+17], [DKL+19]. Let q and n be two integers. Let R, R_q and $R_{\pm 1}$ are the rings of polynomials $\mathbb{Z}[X]/\langle X^n+1\rangle$ over $\mathbb{Z}$, $\mathbb{Z}_q[X]/\langle X^n+1\rangle$ over $\mathbb{Z}_q$ and $\{-1,0,1\}[X]/\langle X^n+1\rangle$ over $\{-1,0,1\}$, respectively. We use an italic small letter, e.g. c, for a polynomial in R. We write a *column* vector in small bold, e.g. $\mathbf{v}$, and a matrix in capital bold, e.g. $\mathbf{A}$. The superscript $(\cdot)^t$ is for the transpose, such as $\mathbf{v}^t$, $\mathbf{A}^t$.

For a positive integer s, $[s]$ means $\{1,2,\cdots,s\}$. For an integer $\alpha > 0$, the centered reduction $r_0 := r \bmod^{\pm}\alpha$ outputs the unique $r_0 \in (-\alpha/2, \alpha/2]$ if α is even, outputs the unique integer $r_0 \in [-(\alpha-1)/2, (\alpha-1)/2]$ if α is odd. By default, an integer w modulo q, i.e., $w \in \mathbb{Z}_q$, is always reduced via the centered reduction; that is $w \bmod^{\pm} q = w$. The reduction $r_1 := r \bmod^{+}\alpha$ returns the unique integer $r_1 \in [0,\alpha)$ such that $r_1 = r \pmod{\alpha}$. We define $\|w\|_\infty$ as $|w \bmod {}^{\pm}q|$ and as $|w|$ for $w \in \mathbb{Z}_q$ and for $w \in \mathbb{Z}$, respectively. For a vector $\mathbf{v}$, the Euclidean norm, the infinity norm and the 1-norm of $\mathbf{v}$ are denoted by $\|\mathbf{v}\|$, $\|\mathbf{v}\|_1$, $\|\mathbf{v}\|_\infty$, respectively. For a polynomial $c \in R$, we abuse the notation $\|c\| := \|\mathbf{c}\|$, $\|c\|_\infty := \|\mathbf{c}\|_\infty$, and $\|c\|_1 := \|\mathbf{c}\|_1$, where $\mathbf{c}$ is the coefficient vector of c. The number of non-zero coefficients, i.e. ± 1's, of a polynomial $c \in R_{\pm 1}$ is denoted by $\#(c)$. For $r > 0$ an integer, we write S_r for the set $\{c \in R : \|c\|_\infty \le r\}$, B_r for the set $\{c \in R_{\pm 1} : \#(c) = r\}$. Then, the cardinality of B_r is $|B_r| = 2^r \cdot \binom{n}{r}$. The notation $\mathcal{A} \Rightarrow 1$ means that the algorithm $\mathcal{A}$ succeeds. We write $a \leftarrow \mathcal{A}^{\mathcal{B},\mathcal{C}}$ to say that the algorithm $\mathcal{A}$ calls or interacts with other algorithms $\mathcal{B}$, $\mathcal{C}$ and finally gets the result a.

B. Lattices and Hardness Assumptions

A *lattice* in $\mathbb{R}^n$ is defined as the set $\mathcal{L}(\mathbf{A}) := \{\sum_{i\in[n]} \mathbf{a}_i z_i : z_i \in \mathbb{Z}\}$ where $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_n] \in \mathbb{R}^{n\times n}$ of n linearly independent vectors is called a *basis* of $\mathcal{L}(\mathbf{A})$. One of the most fundamental problems with respect to lattices is the *shortest vector problem* (SVP) which requires to find the non-zero lattice vectors of shortest norm in a lattice given its basis.

For a random matrix $\mathbf{M} \leftarrow_{\$} \mathbb{Z}^{n\times m}$,

$$\Lambda_q^{\perp}(\mathbf{M}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{M}\mathbf{z} = \mathbf{0} \pmod{q}\},$$

$$\Lambda_q(\mathbf{M}) = \{\mathbf{u} \in \mathbb{Z}^m : \mathbf{u} = \mathbf{M}^t\mathbf{z} \pmod{q} \text{ for some } \mathbf{z} \in \mathbb{Z}^n\}$$

are lattices, usually mentioned in lattice-based cryptography as q-ary lattices.

The security of our blind signature scheme will be based on the hardness assumption of the module version of learning with errors (MLWE) and the short integer solution problem (MSIS).

Definition 2.1 ($\mathsf{MLWE}_{q,k,l,\chi}$): Let $q \ge 2$, k,l be positive integers, and χ be a probability distribution over R. A distribution $\mathcal{L}_{q,k,l,\chi}$ is defined as follows: Chooses a random matrix $\mathbf{A} \leftarrow_{\$} R_q^{k\times l}$, vectors $\mathbf{s}_1 \leftarrow \chi^l$, $\mathbf{s}_2 \leftarrow \chi^k$ and outputs $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in R_q^k$. The decisional version $\mathsf{dMLWE}_{q,k,l,\chi}$ is to distinguish $(\mathbf{A}, \mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2) \leftarrow \mathcal{L}_{q,k,l,\chi}$ from $(\mathbf{A},\mathbf{t}) \leftarrow_{\$} R_q^{k\times l} \times R_q^k$. The search version $\mathsf{sMLWE}_{q,k,l,\chi}$ requires to recover $\mathbf{s}_1$ and/or $\mathbf{s}_2$ given $(\mathbf{A}, \mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2) \leftarrow \mathcal{L}_{q,k,l,\chi}$.

Definition 2.2 ($\mathsf{MSIS}_{q,k,l,\nu}$): Let $q \ge 2$, k,l be positive integers. Given a random matrix $\mathbf{A} \leftarrow_{\$} R_q^{k\times l}$, $\mathbf{u} \leftarrow_{\$} R_q^k$ and a positive real number ν, the (inhomogeneous) short integer problem $\mathsf{MSIS}_{q,k,l,\nu}$ is to find a vector $\mathbf{x} \in R^{k+l}$ such that $[\mathbf{A}\|\mathbf{I}_k]\cdot\mathbf{x} = \mathbf{u} \pmod{q}$ and $\|\mathbf{x}\|_\infty \le \nu$. The homogeneous of MSIS is defined with $\mathbf{u} = \mathbf{0}$ and $\mathbf{x} \ne \mathbf{0}$.

In theory, the hardness of MLWE and MSIS is shown in [LS15] saying that these average-case problems are at least as hard as standard lattice problems restricted to module lattices. Also, [LS15] claims that there is converse reduction from the worst-case to the average-case for these module lattice problem.

C. Commitment Function: Hiding and Binding Properties

A commitment function $\mathsf{com} : \{0,1\}^* \times \{0,1\}^\lambda \to \{0,1\}^\lambda$, $(\mu, \mathbf{d}) \mapsto C$, maps a pair of strings to a commitment string. In this work, we assume that com is *statistically hiding* and *computationally binding*; see [Rüc10] for more details.

D. Blind Signature Schemes and Security Games

A blind signature $\mathsf{BS} := \{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}\}$ consists of the following algorithms generally described as follows:

- $\mathsf{par} \leftarrow \mathsf{Setup}(1^\lambda)$: $\mathsf{Setup}(1^\lambda)$ is a probabilistic polynomial-time (PPT) algorithm that on input a security parameter λ generates a set of system parameters par.
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{par})$: This is a PPT algorithm that takes as input a set of system parameter par to output a public/secret key-pair $(\mathsf{pk}, \mathsf{sk})$.

- $(\Sigma, \mathcal{V}) \leftarrow \mathsf{Sign}(\mathsf{par}, \mathsf{sk}, \mathsf{pk}, \mu)$: Sign is an interaction protocol initiated by a user, say $\mathcal{U}(\mathsf{pk}, \mu)$, and a signer, say $\mathcal{S}(\mathsf{sk})$. In this protocol, the user blinds the message μ to have a blinded message μ^* then the signer is only allowed to sign on μ^*. At the output phase, the user gets the signature Σ for the original message μ while the signer only obtains a view $\mathcal{V}$. It also might that $\Sigma = \perp$, a failure symbol if interaction is not successful.
- $1/0 := \mathsf{Verify}(\mathsf{par}, \mu, \Sigma, \mathsf{pk})$: On input the parameter set par, the public key pk, the message μ and the signature Σ, Verify checks some specific conditions before outputting 1 if all conditions are fulfilled and 0 otherwise.

Correctness: BS is called *correct* if for any $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{par})$, and $(\Sigma, \mathcal{V}) \leftarrow \mathsf{Sign}(\mathsf{par}, \mathsf{sk}, \mathsf{pk}, \mu)$, we have

$$\Pr[\mathsf{Verify}(\mathsf{par}, \mu, \Sigma, \mathsf{pk}) = 1] = 1.$$

For security, a blind signature scheme should be *blind* and *one-more unforgeable*. We formally define these properties in Definition 2.3 and Definition 2.4 through the games $\mathsf{Blind}_{\mathsf{BS}}^{\mathcal{S}^*}$ and $\mathsf{OMUF}_{\mathsf{BS}}^{\mathcal{U}^*}$ (Figure 1, Figure 2) below.

Definition 2.3 (Blindness): BS is *blind* if for any efficient algorithm $\mathcal{S}^*$, $\Pr[\mathsf{Blind}_{\mathsf{BS}}^{\mathcal{S}^*} \Rightarrow 1] - 1/2 \leq \mathsf{negl}(\lambda)$. Particularly, BS is called *perfectly blind* whenever $\Pr[\mathsf{Blind}_{\mathsf{BS}}^{\mathcal{S}^*} \Rightarrow 1] = 1/2$.

Definition 2.4 (One-more Unforgeability): BS is one-more unforgeable if for any efficient algorithm $\mathcal{U}^*$, $\Pr[\mathsf{OMUF}_{\mathsf{BS}}^{\mathcal{U}^*} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$.

```
GAME $\mathsf{Blind}_{\mathsf{BS}}^{\mathcal{S}^*}$ :
1. $\mathsf{par} \leftarrow_{\mathcal{S}^*}^{\mathsf{Setup}(1^\lambda)}$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\mathcal{S}^*}^{\mathsf{KeyGen}(\mathsf{par})}$
2. $(\mu_0, \mu_1) \leftarrow \mathcal{S}^*(\mathsf{pk}, \mathsf{sk})$    // $\mathcal{S}^*$ chooses messages $\mu_0, \mu_1$
3. $b \leftarrow_\$ \{0, 1\}$, $\mathcal{U}_1 := \mathcal{U}(\mathsf{pk}, \mu_b)$, $\mathcal{U}_2 := \mathcal{U}(\mathsf{pk}, \mu_{1-b})$
4. $(\Sigma_1, \Sigma_2) \leftarrow \mathcal{S}^{*\mathcal{U}_1, \mathcal{U}_2}$
5. $b' \leftarrow \mathcal{S}^*, b' \in \{0, 1\}$
6. if $b' = b$: return 1; else: return 0
```

Fig. 1: Blindness game for BS

```
GAME $\mathsf{OMUF}_{\mathsf{BS}}^{\mathcal{U}^*}$ :
1. $\mathsf{par} \leftarrow_\$ \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KeyGen}(\mathsf{par})$
3. $(\mu_1, \Sigma_1), \cdots, (\mu_s, \Sigma_s) \leftarrow \mathcal{U}^{*H, \mathcal{S}(\mathsf{sk})}(\mathsf{pk})$ after at most
   $s - 1$ successful signing interactions with $\mathcal{S}(\mathsf{sk})$
4. if $\mathsf{Verify}(\mathsf{par}, \mu_i, \Sigma_i, \mathsf{pk}) = 1, \forall i \in [s]$
   and $\mu_i \neq \mu_j$ for all $i, j \in [s], i \neq j$: return 1;
   else: return 0
```

Fig. 2: One-more unforgeability game for BS

E. *Dilithium and Fundamental Algorithms*

$\mathsf{Dilithium}$ is a compact and secure signature scheme using uniform sampling and rejection sampling inspired by the framework of the "Fiat-Shamir with Aborts" technique proposed by Lyubashevsky [Lyu09]. Due to limitation of space, we refer to [DLL+17] for more details about the

```
Algorithm 1 MBS.Setup($1^\lambda$)
Input: Security parameter $\lambda$
Output: The set of system parameters and system functions $\mathsf{par} = \{n, k, l, d, q, \gamma_1, \gamma_2, \alpha, \beta, \eta, \sigma$, $\mathsf{Sam}$, $\mathsf{UseHint}_q$, $\mathsf{MakeHint}_q$, $\mathsf{Decompose}_q$, $\mathsf{Power2Round}_q$, $\mathsf{HighBits}_q$, $\mathsf{LowBits}_q$, cryptographic hash function $H\}$
```

```
Algorithm 2 MBS.KeyGen(par)
Input: $\mathsf{par} = \{n, k, l, d, q, \gamma_1, \gamma_2, \alpha, \beta, \eta, \sigma$, $\mathsf{Sam}$, $\mathsf{UseHint}_q$, $\mathsf{MakeHint}_q$, $\mathsf{Decompose}_q$, $\mathsf{Power2Round}_q$, $\mathsf{HighBits}_q$, $\mathsf{LowBits}_q$, cryptographic hash function $H\}$
Output: A key pair $(\mathsf{pk}, \mathsf{sk})$
1: $\rho, \rho' \leftarrow_\$ \{0, 1\}^n$
2: $\mathbf{A} := \mathsf{Sam}(\rho) \in R_q^{k \times l}$
3: $\mathbf{s}_1, \mathbf{s}_2 := \mathsf{Sam}(\rho') \in S_\eta^l \times S_\eta^k$
4: $\mathbf{t} := \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in R_q^k$
5: $\mathbf{t}_1 = \mathsf{Power2Round}_q(\mathbf{t}, d)$, i.e., $\mathbf{t} = \mathbf{t}_1 2^d + \mathbf{t}_0 \pmod q$
6: return Public key $\mathsf{pk} := (\rho, \mathbf{t}_1)$ and secret key $\mathsf{sk} := (\rho, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$
```

algorithms Sam, $\mathsf{Power2Round}_q$, $\mathsf{MakeHint}_q$, $\mathsf{UseHint}_q$, $\mathsf{Decompose}_q$, $\mathsf{HighBits}_q$ and $\mathsf{LowBits}_q$.

Now, we review some useful lemmas in the $\mathsf{Dilithium}$ paper [DKL+19]. Corollary 1 implied from Lemma 2.1 will be useful for the correctness of our proposed blind signature scheme.

Lemma 2.1 (Lemma 1 in [DKL+19]): Let q and α be positive integers such that $q > 2\alpha$, $q = 1 \pmod \alpha$ and α even. Suppose that $\mathbf{r}$ and $\mathbf{z}$ are vectors of integers in $\mathbb{R}_q$ where $\|\mathbf{z}\|_\infty \leq \frac{\alpha}{2}$. Let $\mathbf{h}$ and $\mathbf{h}'$ be bit vectors. Then

i. $\mathsf{UseHint}_q(\mathsf{MakeHint}_q(\mathbf{z}, \mathbf{r}, \alpha), \mathbf{r}, \alpha) = \mathsf{HighBits}_q(\mathbf{r} + \mathbf{z}, \alpha)$.

ii. If $\mathbf{v}_1 = \mathsf{UseHint}_q(\mathbf{h}, \mathbf{r}, \alpha)$, then $\|\mathbf{r} - \mathbf{v}_1\alpha\|_\infty \leq \alpha + 1$. Furthermore, if the Hamming weight of $\mathbf{h}$ is ω then all but at most ω components of $\mathbf{r} - \mathbf{v}_1\alpha$ have magnitude of at most $\alpha/2$ after being centered reduction modulo q.

iii. For any $\mathbf{h}$ and $\mathbf{h}'$, if $\mathsf{UseHint}_q(\mathbf{h}, \mathbf{r}, \alpha) = \mathsf{UseHint}_q(\mathbf{h}', \mathbf{r}, \alpha)$ then $\mathbf{h} = \mathbf{h}'$.

Corollary 1: Let q and α be positive integers such that $q > 2\alpha$, $q = 1 \pmod \alpha$ and α even. Suppose that $\mathbf{r}$, $\mathbf{z}$, $\mathbf{u}$ are vectors of integers in $\mathbb{R}_q$ where $\|\mathbf{z}\|_\infty \leq \frac{\alpha}{2}$. Then

$$\mathsf{UseHint}_q(\mathsf{MakeHint}_q(\mathbf{z}, \mathbf{r}, \alpha), \mathbf{u} \cdot \alpha + \mathbf{r}, \alpha)$$
$$= \mathbf{u} + \mathsf{HighBits}_q(\mathbf{r} + \mathbf{z}, \alpha).$$

Lemma 2.2 (Lemma 2 in [DKL+19]):
If $\|\mathbf{s}\|_\infty \leq \beta$ and $\|\mathsf{LowBits}_q(\mathbf{r}, \alpha)\|_\infty \leq \frac{\alpha}{2} - \beta$ then

$$\mathsf{HighBits}_q(\mathbf{r}, \alpha) = \mathsf{HighBits}_q(\mathbf{r} + \mathbf{s}, \alpha).$$

III. MBS: A Blind Signature Scheme from Dilithium

Our proposed blind signature scheme MBS is constructed via the framework of $\mathsf{Dilithium}$ presented in [DLL+17], [DKL+19].

The algorithms MBS.Setup, MBS.KeyGen, MBS.Sign and MBS.Verify of MBS are detailed in Algorithms 1-3 and Figure 3. The scheme MBS works as follows: It gets started with MBS.Setup which takes a security parameter λ as input and then output a set par of system parameters $n, k, l, d, q, \gamma_1, \gamma_2, \alpha, \beta, \eta, \sigma,$, system functions Sam, $\mathsf{UseHint}_q$, $\mathsf{MakeHint}_q$, $\mathsf{Decompose}_q$, $\mathsf{Power2Round}_q$, $\mathsf{HighBits}_q$, $\mathsf{LowBits}_q$, and also a cryptographic hash function H. System functions are deployed to compress the size of public key and the size of signature in the same way as Dilithium. Afterwards, secret keys and public keys are generated by MBS.KeyGen using par via the same way as in Dilithium. The main algorithm MBS.Sign is an interactive signing protocol between a signer and a user in which the signer performs Phase 1, Phase 3 and Phase 5, while Phase 2 and Phase 4 are done by the user. The processing flow of MBS.Sign is Phase 1 $\rightarrow$ Phase 2 $\rightarrow$ Phase 3 $\rightarrow$ Phase 4 $\rightarrow$ Phase 5. The rejection sampling (aborting) technique is exploited in Step 10, Step 15, Step 17, Step 20 and Step 17 to guarantee the blindness of the proposed scheme. The validity of signature will be then checked using MBS.Verify on input par, the message μ and its corresponding signature Σ. If Σ is valid then MBS.Verify returns 1, otherwise it returns 0.

A. About Hints

The main difference between Dilithium and MBS is related to the blindness of MBS. Concretely, in order for MBS to blind a message and un-blind a blinded signature, we need to use masking factors $\mathbf{a}$ and b (see Step 06 in Phase 2) generated by the user. Note also that the commitment sent from the signer to the user is not $\mathbf{w}$ but $\mathbf{w}_1$, the high bits of $\mathbf{w}$ (see Step 03 and 04 in Phase 1), and the second part of the public key is not $\mathbf{t}$ but $\mathbf{t}_1$, the high bits of $\mathbf{t}$ (see Algorithm 2). Hence we maybe need one more hint, besides the first hint $\mathbf{h}$ coming from the signer, which is similar to Dilithium (see Step 16 in Phase 3). Naturally, the second hint should be from the user due to the use of masking factors. In MBS, the second hint $\mathbf{u}_0$ is given by the user and then packed into result in Step 21 to send to the signer. The hint $\mathbf{u}_0$ is also attached together with the first hint $\mathbf{h}$ into the output signature σ of the user that will be used later in the verification algorithm (see Step 22 in Figure 3 and Algorithm 3).

B. Correctness and Number of Restarts

1) Correctness.: To show the correctness of the MBS scheme, assuming that the signature $\Sigma \neq \perp$, we have to check that $\widehat{\mathbf{u}}_1 = \mathsf{UseHint}_q(\mathbf{h}, \mathbf{Az}^* - c\mathbf{t}_1 2^d - \mathbf{u}_0, 2\gamma_2)$ in Step 2 and Step 3 of Algorithm 3 and $\mathbf{u}_1$ computed in Step 7 of the signing protocol (see Figure 3) are the same. In Step 7 in Figure 3, we have

$$\mathsf{Decompose}_q(\mathbf{w}_1 \cdot 2\gamma_2 + \mathbf{Aa} + b\mathbf{t}_1 2^d, 2\gamma_2) = (\mathbf{u}_1, \mathbf{u}_0).$$

That is, $\mathbf{w}_1 \cdot 2\gamma_2 + \mathbf{Aa} + b\mathbf{t}_1 2^d - \mathbf{u}_0 = \mathbf{u}_1 \cdot 2\gamma_2 \pmod q$. Now we will show that

$$\widehat{\mathbf{u}}_1 = \mathsf{UseHint}_q(\mathbf{h}, \mathbf{Az}^* - c\mathbf{t}_1 2^d - \mathbf{u}_0, 2\gamma_2) = \mathbf{u}_1.$$

Indeed, we have

$$\begin{aligned}
&\mathbf{Az}^* - c\mathbf{t}_1 2^d - \mathbf{u}_0 \\
&= \mathbf{Aa} + \mathbf{w} + b\mathbf{t}_1 2^d - \mathbf{u}_0 - c^*\mathbf{s}_2 + c^*\mathbf{t}_0 \pmod q \\
&= (\mathbf{w}_1 \cdot 2\gamma_2 + \mathbf{Aa} + b\mathbf{t}_1 2^d - \mathbf{u}_0) \\
&\qquad + (\mathbf{w}_0 - c^*\mathbf{s}_2 + c^*\mathbf{t}_0) \pmod q \\
&= \mathbf{u}_1 \cdot 2\gamma_2 + (\mathbf{w}_0 - c^*\mathbf{s}_2 + c^*\mathbf{t}_0) \pmod q.
\end{aligned}$$

It is guaranteed by Step 14 and Step 15 in Figure 3 that $\mathsf{HighBits}_q(\mathbf{w} - c^*\mathbf{s}_2, 2\gamma_2) = \mathbf{w}_1$, and that

$$\|\mathbf{w}_0 - c^*\mathbf{s}_2\|_\infty := \|\mathsf{LowBits}_q(\mathbf{w} - c^*\mathbf{s}_2, 2\gamma_2)\|_\infty < \gamma_2 - \beta,$$

i.e., $\mathsf{HighBits}_q(\mathbf{w}_0 - c^*\mathbf{s}_2, 2\gamma_2) = \mathbf{0}$, the zero vector. With $\mathbf{h} = \mathsf{MakeHint}_q(-c^*\mathbf{t}_0, \mathbf{w}_0 - c^*\mathbf{s}_2 + c^*t_0, 2\gamma_2)$ and $\|c^*\mathbf{t}_0\|_\infty \leq \gamma_2$, using Corollary 1, we obtain

$$\begin{aligned}
&\mathsf{UseHint}_q(\mathbf{h}, \mathbf{u}_1 \cdot 2\gamma_2 + \mathbf{w}_0 - c^*\mathbf{s}_2 + c^*\mathbf{t}_0, 2\gamma_2) \\
&= \mathbf{u}_1 + \mathsf{HighBits}_q(\mathbf{w}_0 - c^*\mathbf{s}_2, 2\gamma_2) = \mathbf{u}_1.
\end{aligned}$$

2) Number of Restarts: We need the following lemma to estimate the probability of restarts that might happen in MBS scheme.

Lemma 3.1 (Adapted from Lemma 3.1 in [Rüc08]): Let n be a sufficient large integer and $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^n$ with arbitrary $\mathbf{a} \in \{\mathbf{v} \in \mathbb{Z}^n : \|\mathbf{v}\|_\infty \leq A\}$ and random $\mathbf{b} \leftarrow_\$ \{\mathbf{v} \in \mathbb{Z}^n : \|\mathbf{v}\|_\infty \leq B\}$. Given $B \geq n \cdot \phi \cdot A$ for $\phi \in \mathbb{N}$ $(\phi > 0)$, we have

$$\Pr_{\mathbf{b}}[\|\mathbf{a} + \mathbf{b}\|_\infty \leq B - A] = \left(1 - \frac{2A}{2B+1}\right)^n \approx e^{-1/\phi}.$$

Now we consider the number of restarts of the proposed scheme.

- *Restarts in Step 10*: They happen locally to guarantee that c^* is independent of c. This ensures the blindness of our scheme. And the restarts in Step 10 follow Lemma 3.1:

$$\Pr_b[\|c+b\|_\infty \leq \sigma - 1] = \left(1 - \frac{2}{2\sigma+1}\right)^n \approx e^{-n/\sigma}. \quad (1)$$

- *Restarts in Step 15:* The same as in [DKL+19, Subsection 3.3], except that we replace c by c^*.
  - The probability of $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$ is still $\left(1 - \frac{\beta}{\gamma_1 - 1/2}\right)^{ln} \approx e^{-n\beta l/\gamma_1}$.
  - The probability of $\|\mathbf{r}_0\|_\infty < \gamma_2 - \beta$ is still $\left(1 - \frac{\beta + 1/2}{\gamma_2}\right)^{kn} \approx e^{-n\beta k/\gamma_2}$.
  - Note that, β is set up such that

  $$\|c^*\mathbf{s}_i\|_\infty \leq \beta, \text{ for } i = 1, 2, \quad (2)$$

  with overwhelming probability (Step 13 in Figure 2) (see also [DLL+17, Section 4.4] to know the role of β and how to choose it). Due to Lemma 2.2, if $\|c^*\mathbf{s}_2\|_\infty < \beta$ and $\|\mathbf{r}_0\|_\infty < \gamma_2 - \beta$ then $\mathbf{r}_1 = \mathbf{w}_1$ with overwhelming probability.

  Thus, the probability that Step 15 passes is

$$\Pr[\text{Step 15 passes}] \approx e^{-n\beta(l/\gamma_1 + k/\gamma_2)}. \quad (3)$$

```
PHASES OF THE SIGNER $\mathcal{S}(\mathsf{par}, \rho, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_1, \mathbf{t}_0)$:

Phase 1:
01. $\mathbf{A} := \mathsf{Sam}(\rho) \in R_q^{k\times l}$
02. $r \leftarrow_\$ \{0,1\}^n, \mathbf{y} := \mathsf{Sam}(r) \in S^l_{\gamma_1 - 1}, \mathbf{w} = \mathbf{A}\mathbf{y} \in R_q^k$
03. $(\mathbf{w}_1, \mathbf{w}_0) = \mathsf{Decompose}_q(\mathbf{w}, 2\gamma_2)$
      (i.e. $\mathbf{w} = 2\gamma_2\mathbf{w}_1 + \mathbf{w}_0 \pmod q$)
04. Send $\mathbf{w}_1$ to the user. The next is Phase 2

Phase 3:
12. $\mathbf{z} = \mathbf{y} + c^*\mathbf{s}_1$
13. $\|c^*\mathbf{s}_i\|_\infty \le \beta$, $i = 1, 2$ with overwhelming probability
14. $(\mathbf{r}_1, \mathbf{r}_0) = \mathsf{Decompose}_q(\mathbf{w} - c^*\mathbf{s}_2, 2\gamma_2)$
      (i.e. $\mathbf{w} - c^*\mathbf{s}_2 = 2\gamma_2\mathbf{r}_1 + \mathbf{r}_0 \pmod q$)
15. if $(\|\mathbf{z}\|_\infty \ge \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \ge \gamma_2 - \beta$ or $\mathbf{r}_1 \ne \mathbf{w}_1)$:
        restart from Phase 1
16. $\mathbf{v}_0 := \mathbf{w}_0 - c^*\mathbf{s}_2 + c^*\mathbf{t}_0$, $\mathbf{h} := \mathsf{MakeHint}_q(-c^*\mathbf{t}_0, \mathbf{v}_0, 2\gamma_2)$
17. if $(\|c^*\mathbf{t}_0\|_\infty \ge \gamma_2)$: restart from Phase 1
18. Send $\mathbf{z}, \mathbf{h}$ to the user. The next is Phase 4

Phase 5:
24. if $(\mathsf{result} \ne \mathsf{accept})$: Parse $\mathsf{result} := (\mathbf{a}, b, c, C, \mathbf{u}_0)$
25.     $\mathbf{u} = \mathbf{A}\mathbf{a} + \mathbf{w}_1 2\gamma_2 + b\mathbf{t}_1 2^d \pmod q$
        $\mathbf{u}_1 = \mathsf{HighBits}_q(\mathbf{u}, 2\gamma_2)$
26.     $\hat{\mathbf{u}} = \mathbf{A}\mathbf{a} + \mathbf{A}\mathbf{z} - c\mathbf{t}_1 2^d - \mathbf{u}_0 \pmod q$
        $\hat{\mathbf{u}}_1 = \mathsf{UseHint}_q(\mathbf{h}, \hat{\mathbf{u}}, 2\gamma_2)$
27.     if $(c^* - b = c = H(\mathbf{u}_1, C)$ and $c = H(\hat{\mathbf{u}}_1, C)$
            and $\|\mathbf{z}^*\|_\infty \ge \alpha + \beta - \gamma_1)$: restart from Phase 1
28. Output: the view $\mathcal{V} = (\mathbf{w}_1, c^*, \mathbf{y}, \mathbf{z})$
```

```
PHASES OF THE USER $\mathcal{U}(\mathsf{par}, \mu, \rho, \mathbf{t}_1)$ :

Phase 2:
05. $\mathbf{A} := \mathsf{Sam}(\rho) \in R_q^{k\times l}$
06. $\mathbf{a} \leftarrow_\$ S^l_\alpha$, $b \leftarrow_\$ S_\sigma$
07. $\mathbf{d} \leftarrow_\$ \{0,1\}^n$, $C := \mathsf{com}(\mu, \mathbf{d})$,
      $\mathbf{u} = \mathbf{A}\mathbf{a} + \mathbf{w}_1 2\gamma_2 + \mathbf{t}_1 b 2^d \pmod q$
      $(\mathbf{u}_1, \mathbf{u}_0) = \mathsf{Decompose}_q(\mathbf{u}, 2\gamma_2)$
      (i.e. $\mathbf{u} = 2\gamma_2\mathbf{u}_1 + \mathbf{u}_0 \pmod q$)
08. $c = H(\mathbf{u}_1, C) \in B_\kappa$, $c^* = c + b$
09. if $(c^* \in S_{\sigma-1})$: output $c^*$
10. else: restart from Phase 2
11. Send $c^*$ back to the signer. The next is Phase 3

Phase 4:
19. $\mathbf{z}^* = \mathbf{z} + \mathbf{a}$
20. if $(\|\mathbf{z}^*\|_\infty < \alpha + \beta - \gamma_1)$ : $\mathsf{result} := \mathsf{accept}$
21. else: $\mathsf{result} := (\mathbf{a}, b, c, C, \mathbf{u}_0)$
22. Output: $(\mu, \Sigma = (\mathbf{z}^*, c, \mathbf{d}, \mathbf{h}, \mathbf{u}_0))$
        or $\perp$ when $\mathsf{result} \ne \mathsf{accept}$
23. Send $\mathsf{result}$ back to the signer. The next is Phase 5
```

Fig. 3: The interactive signing protocol $\mathsf{MBS.Sign}(\mathsf{par}, \mathsf{sk}, \mu, \mathsf{pk})$

- *Restarts in Step 17:* We will set up the parameters such that
$$\|c^*\mathbf{t}_0\|_\infty \le \gamma_2 - 1, \qquad (4)$$
with overwhelming probability
- *Restarts in Step 27 in Phase 5:* We only care about the condition $\|\mathbf{z}^*\|_\infty \ge \alpha + \beta - \gamma_1$ which might be caused by the rejection sampling. Assumed that $\gamma_1 - \beta \ll \alpha$, applying again Lemma 3.1, we have

$$\Pr_b[\|\mathbf{z}^*\|_\infty \le \alpha + \beta - \gamma_1] = \left(1 - \frac{2\gamma_1 - 2\beta + 1}{2\alpha + 1}\right)^{nl} \approx e^{-(\gamma_1 - \beta)nl/\alpha}. \qquad (5)$$

C. Min-entropy and Zero Knowledge Proof

Definition 3.1 (Definition 2.6 in [KLS18]): If the most likely value of a random variable W that is chosen from a discrete distribution D occurs with probability $2^{-\epsilon}$, then we say that $\mathsf{min\text{-}entropy}(W|W \leftarrow D) = \epsilon$.

First, we recall a lemma relating to the min-entropy of Dilithium.

```
Algorithm 3 MBS.Verify$(\mathsf{par}, \mu, \Sigma)$
Input: $\mathsf{par}$, $\mu, \Sigma = (\mathbf{z}^*, c, \mathbf{d}, \mathbf{h}, \mathbf{u}_0)$, $\mathsf{PK} = (\rho, \mathbf{t}_1)$
Output: 1 or 0
1: $\mathbf{A} := \mathsf{Sam}(\rho) \in R_q^{k\times l}$
2: $\hat{\mathbf{u}}_1 = \mathsf{UseHint}_q(\mathbf{h}, \mathbf{A}\mathbf{z}^* - c\mathbf{t}_1 2^d - \mathbf{u}_0, 2\gamma_2)$
3: $c' = H(\hat{\mathbf{u}}_1, \mathbf{com}(\mu, \mathbf{d}))$
4: if $c' = c$ and $\|\mathbf{z}^*\|_\infty \le \alpha + \beta - \gamma_1$ then
5:     return 1
6: else
7:     return 0
8: end if
```

Lemma 3.2 (Lemma C.1 in [KLS18]): Given $\mathbf{A} \leftarrow_\$ R_q^{k\times l}$ and $\mathbf{y} \leftarrow_\$ S^l_{\gamma_1 - 1}$. Then

$$\Pr_{\mathbf{A}}\left[\forall \mathbf{w}_1 : \Pr_{\mathbf{y}}[\mathsf{HighBits}_q(\mathbf{A}\mathbf{y}, 2\gamma_2) = \mathbf{w}_1] \le \left(\frac{2\gamma_2 + 1}{2\gamma_1 - 1}\right)^n\right] > 1 - (1/q)^{kl}. \qquad (6)$$

Proof 1 (of Lemma 3.2):

For a matrix $\mathbf{A} \leftarrow_\$ R_q^{k\times l}$, with probability at least $1 - (1/q)^{kl}$, one of $k \cdot l$ polynomials in the matrix $\mathbf{A}$ is invertible over R_q. Assume that $[A_1, \cdots, A_l]$ is the row of

$\mathbf{A}$ that contains the invertible polynomial, say without loss of generality A_1. Let call T the set of all integers w such that $\mathsf{HighBits}_q(w, 2\gamma_2) = w_1$. Then the size of T is at most $(2\gamma_2+1)^n$ (which is the number of possibility of w_0 such that $(w_1, w_0) = \mathsf{Decompose}_q(w, 2\gamma_2)$, i.e. $w = w_1 2\gamma_2 + w_0 \pmod q$). At this point we have

$$\begin{aligned}&\Pr_{\mathbf{y}}\left[\exists t^* \in T : \sum A_i y_i = t^*\right] \\ &= \Pr_{\mathbf{y}}[y_1 = A_1^{-1}(t^* - \sum A_i y_i)] \\ &\le \left(\frac{2\gamma_2+1}{2\gamma_1-1}\right)^n,\end{aligned}$$

since the size of $S^l_{\gamma_1-1}$ is exactly $(2\gamma_1-1)^n$.

Note that in Step 7 of Figure 3, because $\mathbf{u} := \mathbf{A}\mathbf{a} + \mathbf{w}_1 2\gamma_2 + \mathbf{t}_1 b 2^d$, we have $\mathbf{u}_1 = \mathsf{HighBits}_q(\mathbf{u}, 2\gamma_2) = \mathbf{w}_1 + \mathsf{HighBits}_q(\mathbf{A}\mathbf{a} + \mathbf{t}_1 b 2^d, 2\gamma_2)$. Hence, for the min-entropy of MBS, it suffices to prove the following lemma.

Lemma 3.3 (Min-entropy of MBS): Given $\mathbf{A} \leftarrow_\$ R_q^{k\times l}$, $(\mathbf{s}_1, \mathbf{s}_2) \leftarrow_\$ S^l_\eta \times S^k_\eta$, $\mathbf{t} := \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in R^k_q$, $\mathbf{t}_1 := \mathsf{Power2Round}_q(\mathbf{t}, d)$, $\mathbf{a} \leftarrow_\$ S^l_\alpha$, $b \leftarrow_\$ S_\sigma$, $\mathbf{y} \leftarrow_\$ S^l_{\gamma_1-1}$. Then for all $\bar{\mathbf{w}}_1$,

$$\Pr_{\mathbf{a}\leftarrow_\$ S^l_\alpha, b\leftarrow_\$}[\mathsf{HighBits}_q(\mathbf{A}\mathbf{a}+\mathbf{t}_1 b 2^d, 2\gamma_2) = \bar{\mathbf{w}}_1] \le \left(\frac{2\gamma_2+1}{2\alpha+1}\right)^n. \quad (7)$$

Proof 2 (of Lemma 3.3): The proof here is similar to the one of Lemma 3.2 with noting that by the MLWE assumption, that is, $\mathbf{t}$ behaves as a uniform variable over R^k_q. For all $\bar{w}_1$, define T' to be the set of all integers $\bar{w}$ such that $\mathsf{HighBits}_q(\bar{w}, 2\gamma_2) = \bar{w}_1$. Again, because the size of T' is at most $(2\gamma_2+1)^n$ and the size of S^l_α is exactly $(2\alpha+1)^n$, we have

$$\begin{aligned}&\Pr_{\mathbf{a},b}\left[\exists t^* \in T' : \sum A_i y_i + t_1 b 2^d = t^*\right] \\ &= \Pr_{\mathbf{a}}[a_1 = A_1^{-1}(t^* - \sum A_i a_i - t_1 b 2^d)] \\ &\le \left(\frac{2\gamma_2+1}{2\alpha+1}\right)^n.\end{aligned}$$

The lemmas above (with appropriately chosen parameters) together with an argument of zero knowledge proof that is similar to the one in [DKL+19, Appendix B] enables us to simulate a perfect environment for the security proof performed in Section IV.

D. Size of the Public Key and the Signature

The public key consists of ρ of size $n/8$ bytes and $\mathbf{t}_1$ of size $(nk(\lceil\log q\rceil - d))/8$ bytes, so the size of the public key is totally $(nk(\lceil\log q\rceil - d) + n)/8$. The signature $\Sigma = (\mathbf{z}^*, c, \mathbf{d}, \mathbf{h}, \mathbf{u}_0)$ consists of $\mathbf{z}^*$ which is $nl\lceil\log(2\alpha)\rceil/8$ bytes, c of size $\kappa(\log n + 1)/8$ bytes, $\mathbf{d}$ of size $n/8$ bytes, $\mathbf{h}$ of size $nk/8$ bytes and $\mathbf{u}_0$ of size $nk\lceil\log(2\gamma_2)\rceil/8$. The total size of the signature is

$$(nl\lceil\log(2\alpha)\rceil + \kappa(\log n+1) + n + nk + nk\lceil\log(2\gamma_2)\rceil)/8 \text{ bytes.}$$

IV. Security Analysis of MBS

A. Security against Key Recovery Attacks

The keygen genration algorithm MBS.KeyGen of our scheme is quite similar to the algorithm Gen of Dilithium. Hence, our scheme is secure against key recovery attacks thanks to the MLWE assumption. Also, note that MLWE also guarantees that $(\mathbf{A}, \mathbf{t}) \leftarrow \mathcal{L}_{q,k,l,S_{\eta'}}$ is indistinguishable from uniform over $R_q^{k\times l} \times R_q^k$.

B. Blindness

The blindness of MBS is the direct consequence of the following lemma.

Lemma 4.1 (Adapted from Lemma 3.4 in [Rüc08]): Let $k \in \mathbb{N}$, $\mathbf{a}, \mathbf{a}', \mathbf{b} \in \mathbb{Z}^k$ with arbitrary $\mathbf{a}, \mathbf{a}' \in \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \le A\}$ and random $\mathbf{b} \leftarrow_\$ \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \le B\}$ for $B > A$. We define two random variables $\mathbf{c} \leftarrow \mathbf{a} + \mathbf{b}$ and $\mathbf{c}' \leftarrow \mathbf{a}' + \mathbf{b}$ if $\max\{\|\mathbf{a}+\mathbf{b}\|_\infty, \|\mathbf{a}'+\mathbf{b}\|_\infty\} \le B - A$, otherwise we re-sample $\mathbf{b}$. Then $\Delta(\mathbf{c}, \mathbf{c}') = 0$.

Theorem 4.2 (Blindness): Our MBS scheme is blind provided that the commitment function com is hiding.

Proof 3 (of Theorem 4.2): It is easy to see that the blindness of our BRS scheme is guaranteed by Lemma 3.1 and Lemma 4.1 with appropriately chosen parameters, assuming the hiding property of the commitment com to be satisfied.

C. One-more Unforgeability

The one-more unforgeability for the MBS scheme is based on the intractability of the MSIS assumption. Our proof for this basically follows the proof of [Rüc10, Theorem 3.8] exploiting the rewinding (forking lemma) technique. Concretely, we will show in the following theorem that if there exists a forger who breaks the one-more unforgeability of MBS then one can build an efficient MSIS solver.

Theorem 4.3 (One-more Unforgeability): Provided that the commitment function com used in the MBS scheme is binding. Assume that there is a forger $\mathcal{U}^*$ breaks the one-more unforgeablity of MBS, The attacker makes at most h queries to the hash function, and at most $s-1$ queries to the signing protocol. Let θ be the probability of a restart occuring in the MBS scheme. Let ψ non-negligible be the success probability of the attacker. Then we can construct a polynomial-time solver $\mathcal{G}$ that can find a solution to the $\mathsf{MSIS}_{q,k,l,\nu}$ instance with $\nu = \max\{2(\gamma_1-\beta), 6\gamma_2+2+60\cdot 2^d\}$ and probability ψ_{total} not smaller than $\min\{\frac{1}{2s}(1-\theta)\left(1-\frac{1}{|B_\kappa|}\right)\left(\frac{\psi-\frac{1}{|B_\kappa|}}{h} - \frac{1}{|B_\kappa|}\right), \psi\left(1-\frac{1}{|B_\kappa|}\right)\}$.

Proof 4 (of Theorem 4.3):

The MSIS solver $\mathcal{G}$ is constructed using $\mathcal{U}^*$ as a subroutine as follows:

1) **Setup.** $\mathcal{G}$ chooses a security parameter λ, then generates the system parameters and the system functions included in par, the public key/secret key $\mathsf{PK} := (\rho, \mathbf{t}_1)$, $\mathsf{SK} := (\rho, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$ by calling $\mathsf{Setup}(1^\lambda)$ and then invoking $\mathsf{KeyGen}(\mathsf{par})$. Next, $\mathcal{G}$ sends par and PK to $\mathcal{U}^*$. Note that, both $\mathcal{G}$ and $\mathcal{U}^*$ can easily recover the

matrix $\mathbf{A} := \mathsf{Sam}(\rho) \in R_q^{k\times l}$ using Sam from ρ and $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \pmod q = \mathbf{t}_1 2^d + \mathbf{t}_0 \pmod q, \mathbf{t} \in R_q^k$. Remember that, $\mathcal{G}$ keeps SK secret. The goal of $\mathcal{G}$ is solving the following MSIS problem which is hard:

$$\begin{aligned}&\text{Find } \mathbf{x} \in R^{k+l+1} \text{ such that } [\mathbf{A}\|\mathbf{I}_k\|\mathbf{t}]\cdot\mathbf{x} = \mathbf{0} \pmod q,\\ &\text{subject to } \|\mathbf{x}\| \le \nu.\end{aligned} \tag{8}$$

The solver $\mathcal{G}$ initials and keeps a list *List* $= ((\mathbf{u}_1, C), c) \subset R_q^k \times \{0,1\}^n \times B_\kappa$ containing of random oracle queries $(\mathbf{u}_1, C) \leftarrow_\$ R_q^k \times \{0,1\}^n$ and their corresponding hash value $c \in B_\kappa$. Also, $\mathcal{G}$ creates a list $\mathcal{R} := \{r_1, \cdots, r_h\}$ where r_i's are possible replies of the random oracle sampled uniformly at random fromB_κ as a set of replies of H and also chooses a random tape rt. The solver $\mathcal{G}$ runs $\mathcal{U}^*(\mathsf{par}, \mathsf{PK}, \mathsf{rt})$ as a black-box routine in the following two phases:

2) **Queries.** The forger $\mathcal{U}^*$ makes at most h hash queries and makes at most $s-1$ signing queries in an adaptive manner. If $\mathcal{U}^*$ makes the hash query $(\mathbf{u}_1, C)$, $\mathcal{G}$ will check the existence of the hash query in *List*. If $(\mathbf{u}_1, C)$ is already in *List*, $\mathcal{G}$ sends the corresponding hash value c to the forger $\mathcal{U}^*$. If $(\mathbf{u}_1, C)$ is not in *List* yet, $\mathcal{G}$ assigns $c := r_i$, where r_i is the unused value that $\mathcal{G}$ sees first in $\mathcal{R}$. Then $\mathcal{G}$ updates the list *List* by adding the query-hash value pair $((\mathbf{u}_1, C), c)$ into it and finally sends $\mathcal{U}^*$ the value c. In the reduction, $\mathcal{U}^*$ acts as the (dishonest) user and the solver $\mathcal{G}$ acts as the signer, they initiates at most $s-1$ signing interactions following the real $\mathsf{MBS.Sign}$ in Figure 3.
3) **Output.** With non-negligible probability ψ, the forger $\mathcal{U}^*$ eventually succeeds to obtain s message/signature pairs
$$(\mu_1, (\mathbf{z}_1^*, c_1, \mathbf{d}_1, \mathbf{h}_1, \mathbf{u}_{0,1}), \cdots, (\mu_s, (\mathbf{z}_s^*, c_s, \mathbf{d}_s, \mathbf{h}_s, \mathbf{u}_{0,s}),$$
only making at most $s-1$ signing interactions, where $\mu_i \ne \mu_j$ for all $i, j \in [s], i \ne j$. At this stage, the solver $\mathcal{G}$ selects uniformly at random an index $\tau \in [s]$ such that $c_\tau = r_{i_0}$ for some $i_0 \in [h]$.

Now, the solver $\mathcal{G}$ exploits the rewinding technique: He again samples random oracle answers $\{r_i', \cdots, r_h'\}$ from B_κ, creates $\mathcal{R}' := \{r_1, \cdots, r_{i-1}, r_i', \cdots, r_h'\}$ and then invokes $\mathcal{U}^*(\mathsf{par}, \mathsf{PK}, \mathsf{rt})$ using $\mathcal{R}'$ instead of $\mathcal{R}$. Again, after at most $s-1$ signing interactions, $\mathcal{U}^*$ outputs $(\mu_\tau', (\mathbf{z'}_\tau^*, c_\tau', \mathbf{d}_\tau', \mathbf{h}_\tau', \mathbf{u}_{0,\tau}'))$ among $s-1$ other pairs. If $c_\tau \ne c_\tau'$, $\mathcal{G}$ uses

$$(\mu_\tau, (\mathbf{z}_\tau^*, c_\tau, \mathbf{d}_\tau, \mathbf{h}_\tau, \mathbf{u}_{0,\tau})) \text{ and } (\mu_\tau', (\mathbf{z'}_\tau^*, c_\tau', \mathbf{d}_\tau', \mathbf{h}_\tau', \mathbf{u}_{0,\tau}')), \tag{9}$$

aiming at extracting a solution to the MSIS problem (8). If $c_\tau = c_\tau'$, the solver $\mathcal{G}$ reruns $\mathcal{U}^*(\mathsf{par}, \mathsf{PK}, \mathsf{rt}')$ at most h^s times with a different random tape rt'.

Analysis. Below, we will analysis to see how $\mathcal{G}$ extracts a solution to the MSIS problem (8) from the outputs of $\mathcal{U}^*$.

First of all, we can see that the environment of $\mathcal{U}^*$ is perfectly modeled by $\mathcal{G}$ even with restarts which happen with probability θ as in real signing protocol $\mathsf{MBS.Sign}$. Due to having at most $s-1$ real interactions performed between $\mathcal{G}$ and $\mathcal{U}^*$, at least one of s output signatures is not from these real interactions. For the index $\tau \in [s]$ of the forged signature $(\mathbf{z}_\tau^*, c_\tau, \mathbf{d}_\tau, \mathbf{h}_\tau, \mathbf{u}_{0,\tau})$ on μ_τ, the success probability of $\mathcal{G}$ in guessing correctly τ is at least $1/s$. The probability that c_τ is correctly guessed is exactly $1/|B_\kappa|$ since it is chosen uniformly at random. Regarding reruns of $\mathcal{U}^*$, with probability $1/2$, at least one rerun gives the same pair (i_0, τ) fulfilling that $r_{i_0} = c_\tau$ as the first run of $\mathcal{U}^*$.

When $\mathcal{G}$ invokes the rewinding technique, by the forking lemma [Rüc08, Lemma A.1], $\mathcal{U}^*$ breaks the one-more unforgeability again and outputs $(\mathbf{z'}_\tau^*, c_\tau', \mathbf{d}_\tau', \mathbf{h}_\tau', \mathbf{u}_{0,\tau}')$ on some μ_τ', with probability

$$\psi_{\mathsf{fork}} \ge (1-\theta)(\psi - 1/|B_\kappa|)((\psi - 1/|B_\kappa|)/h - 1/|B_\kappa|),$$

taking restarts which may happen with probability at most θ into account, using the same random oracle query, say $(\mathbf{u}_{1,\tau}, C_\tau)$, as in the first run. The random oracle query used, $(\mathbf{u}_{1,\tau}, C_\tau)$, is exactly recovered from the signatures in (9) by

$$\begin{aligned}&(\mathsf{UseHint}_q(\mathbf{h}_\tau, \widehat{\mathbf{u}}_\tau, 2\gamma_2), \mathsf{com}(\mu_\tau, \mathbf{d}_\tau))\\ &= (\mathsf{UseHint}_q(\mathbf{h}_\tau', \widehat{\mathbf{u}}_\tau', 2\gamma_2), \mathsf{com}(\mu_\tau', \mathbf{d}_\tau')),\end{aligned}$$

where $\widehat{\mathbf{u}}_\tau = \mathbf{A}\mathbf{z}_\tau^* - c_\tau \mathbf{t}_1 2^d - \mathbf{u}_{0,\tau}$, $\widehat{\mathbf{u}}_\tau' = \mathbf{A}\mathbf{z'}_\tau^* - c_\tau' \mathbf{t}_1 2^d - \mathbf{u}_{0,\tau}'$ and $C_\tau = \mathsf{com}(\mu_\tau, \mathbf{d}_\tau) = \mathsf{com}(\mu_\tau', \mathbf{d}_\tau')$. According to the binding property of com, it must be that $\mu_\tau = \mu_\tau'$ and $\mathbf{d}_\tau = \mathbf{d}_\tau'$, so we do not care about them. Since then, we have that

$$\mathsf{UseHint}_q(\mathbf{h}_\tau, \widehat{\mathbf{u}}_\tau, 2\gamma_2) = \mathsf{UseHint}_q(\mathbf{h}_\tau', \widehat{\mathbf{u}}_\tau', 2\gamma_2).$$

hence,

$$2\gamma_2 \cdot \mathsf{UseHint}_q(\mathbf{h}_\tau, \widehat{\mathbf{u}}_\tau, 2\gamma_2) = 2\gamma_2 \cdot \mathsf{UseHint}_q(\mathbf{h}_\tau', \widehat{\mathbf{u}}_\tau', 2\gamma_2),$$

which is rewritten as

$$\mathbf{A}\mathbf{z}_\tau^* - c_\tau \mathbf{t}_1 2^d - \mathbf{u}_{0,\tau} + \bar{\mathbf{u}}_\tau = \mathbf{A}\mathbf{z'}_\tau^* - c_\tau' \mathbf{t}_1 2^d - \mathbf{u}_{0,\tau}' + \bar{\mathbf{u}}_\tau', \tag{10}$$

with $\|\bar{\mathbf{u}}_\tau\|_\infty, \|\bar{\mathbf{u}}_\tau'\|_\infty \le 2\gamma_2 + 1$ by Lemma 2.1(ii). Equation (10) is equivalent to

$$\mathbf{A}(\mathbf{z}_\tau^* - \mathbf{z'}_\tau^*) + (c_\tau' - c_\tau)\mathbf{t}_1 2^d = \mathbf{u}_{0,\tau} - \bar{\mathbf{u}}_\tau - \mathbf{u}_{0,\tau}' + \bar{\mathbf{u}}_\tau',$$

which is

$$\mathbf{A}(\mathbf{z}_\tau^* - \mathbf{z'}_\tau^*) + (c_\tau' - c_\tau)\mathbf{t} - \mathbf{u}_{0,\tau} + \mathbf{u}_{0,\tau}' - \bar{\mathbf{u}}_\tau' + \bar{\mathbf{u}}_\tau - (c_\tau' - c_\tau)\mathbf{t}_0 = \mathbf{0},$$

or in the form

$$[\mathbf{A}\|\mathbf{I}_k\|\mathbf{t}]\cdot\mathbf{x} = \mathbf{0},$$

where $\mathbf{x} = (\mathbf{z}_\tau^* - \mathbf{z'}_\tau^*, \bar{\mathbf{x}}, c_\tau' - c_\tau)$, with $\bar{\mathbf{x}} = \mathbf{u}_{0,\tau}' - \mathbf{u}_{0,\tau} - \bar{\mathbf{u}}_\tau' + \bar{\mathbf{u}}_\tau - (c_\tau' - c_\tau)\mathbf{t}_0$.

Note that, $\|\mathbf{z}_\tau^* - \mathbf{z'}_\tau^*\|_\infty \le 2(\alpha + \beta - \gamma_1)$, and

$$\begin{aligned}\|\bar{\mathbf{x}}\|_\infty &\le \|\mathbf{u}_{0,\tau}' - \mathbf{u}_{0,\tau}\|_\infty + \|\bar{\mathbf{u}}_\tau' - \bar{\mathbf{u}}_\tau\|_\infty + \|c_\tau' - c_\tau\|_1 \cdot \|\mathbf{t}_0\|_\infty\\ &\le 6\gamma_2 + 2 + 60\cdot 2^d.\end{aligned}$$

Hence, $\|\mathbf{x}\|_\infty \le \max\{2(\alpha+\beta-\gamma_1), 6\gamma_2 + 2 + 60\cdot 2^d\}$. Again, $\mathbf{x} \ne \mathbf{0}$ as $c_\tau \ne c_\tau'$.

To summarize, we have the success probability in solving MSIS (8) is at least

$$\begin{aligned}\psi_{\mathsf{sol}} &\ge \frac{1}{2s}\psi_{\mathsf{fork}}\\ &\ge \frac{1}{2s}(1-\theta)(\psi - 1/|B_\kappa|)((\psi - 1/|B_\kappa|)/h - 1/|B_\kappa|).\end{aligned}$$

Now we assume that there is an abort happening in Phase 5. Then we will show that a cheating user cannot output a valid signature without solving the MSIS problem. To trigger a restart, after Phase 4, the $\mathsf{result} := (\mathbf{a}, b, c, C, \mathbf{u}_0)$ is sent to the signer. Together with the view of the signer $\mathcal{V} = (\mathbf{w}_1, c^*, \mathbf{y}, \mathbf{z})$, the signer computes $\mathbf{u} = \mathbf{A}\mathbf{a} + \mathbf{w}_1 2\gamma_2 + b\mathbf{t}_1 2^d \pmod q$, $\mathbf{u}_1 = \mathsf{HighBits}_q(\mathbf{u}, 2\gamma_2)$, $\widehat{\mathbf{u}} = \mathbf{A}\mathbf{a} + \mathbf{A}\mathbf{z} - c\mathbf{t}_1 2^d - \mathbf{u}_0 \pmod q$, $\widehat{\mathbf{u}}_1 = \mathsf{UseHint}_q(\mathbf{h}, \widehat{\mathbf{u}}, 2\gamma_2)$. and check all following conditions:

$$c^* - b = c = H(\mathbf{u}_1, C) \tag{11}$$

$$c = H(\widehat{\mathbf{u}}_1, C) \tag{12}$$

$$\|\mathbf{z} + \mathbf{a}\|_\infty > \alpha + \beta - \gamma_1 \tag{13}$$

If all conditions are satisfied, the signer will restart the protocol. Assume that the user can get from this abort a valid signature $\Sigma = (\mathbf{z}'^*, c', \mathbf{d}', \mathbf{h}', \mathbf{u}'_0)$ with $\|\mathbf{z}'^*\|_\infty \le \alpha + \beta - \gamma_1$. Set $\widehat{\mathbf{u}}' = \mathbf{A}\mathbf{z}'^* - c'\mathbf{t}_1 2^d - \mathbf{u}'_0 \pmod q$, $\widehat{\mathbf{u}}'_1 = \mathsf{UseHint}_q(\mathbf{h}, \widehat{\mathbf{u}}', 2\gamma_2)$. In case $c' = c$, we have by (12) that $\widehat{\mathbf{u}}_1 = \widehat{\mathbf{u}}'_1$, i.e.,

$$\mathsf{UseHint}_q(\mathbf{h}, \widehat{\mathbf{u}}, 2\gamma_2) = \mathsf{UseHint}_q(\mathbf{h}, \widehat{\mathbf{u}}', 2\gamma_2).$$

That is, $\mathbf{A}\mathbf{z} + \mathbf{A}\mathbf{a} - \mathbf{u}_0 + \bar{\mathbf{u}} = \mathbf{A}\mathbf{z}'^* - \mathbf{u}'_0 + \bar{\mathbf{u}}'$, equivalently,

$$\mathbf{A}(\mathbf{z} + \mathbf{a} - \mathbf{z}'^*) + \mathbf{u}'_0 - \mathbf{u}_0 + \bar{\mathbf{u}} - \bar{\mathbf{u}}' = \mathbf{0},$$

where $\|\bar{\mathbf{u}}\|_\infty \le 2\gamma_2 + 1$ and $\|\bar{\mathbf{u}}'\|_\infty \le 2\gamma_2 + 1$. Define

$$\mathbf{x} := (\mathbf{z} + \mathbf{a} - \mathbf{z}'^*, \mathbf{u}'_0 - \mathbf{u}_0 + \bar{\mathbf{u}} - \bar{\mathbf{u}}', \mathbf{0}),$$

we have $[\mathbf{A}\|\mathbf{I}_k, \mathbf{t}] \cdot \mathbf{x} = \mathbf{0} \pmod q$. If $\mathbf{x} = 0$, then we have $\mathbf{z} + \mathbf{a} = \mathbf{z}'^*$, hence $\|\mathbf{z} + \mathbf{a}\|_\infty \le \alpha + \beta - \gamma_1$ which contradicts with (13). Hence, we have $\mathbf{x} \neq \mathbf{0}$ and $\|\mathbf{x}\|_\infty \le \max\{2(\gamma_1 - \beta), 6\gamma_2 + 2\}$. In case $c' \neq c$, we assume that the cheating user might be successful by hiding c' in some c^*. By (11), there exists some $b' \neq b$ such that $c^* = c + b = c' + b'$. in this case, the user had to be able to predict what H outputted in order to compute b'. And the success probability of the prediction is exactly $1 - 1/|B_\kappa|$. Therefore, the success probability of $\mathcal{G}$ in solving the MSIS problem (8) when an abort happens is at least $\psi_{\mathsf{abort}} \ge \psi(1 - 1/|B_\kappa|)$.

To conclude, with the overall success probability $\psi_{\mathsf{total}} \ge \min\{\psi_{\mathsf{sol}}, \psi_{\mathsf{abort}}\}$, the solver $\mathcal{G}$ can solve the $\mathsf{MSIS}_{q,k,l,\nu}$ problem (8): $[\mathbf{A}\|\mathbf{I}_k\|\mathbf{t}] \cdot \mathbf{x} = \mathbf{0} \pmod q$, where $\|\mathbf{x}\|_\infty \le \nu := \max\{2(\gamma_1 - \beta), 6\gamma_2 + 2 + 60 \cdot 2^d\}$. Clearly, ψ_{total} is non-negligible if ψ is non-negligible.

V. Choosing Parameters for MBS

Compared with Dilithium, two more parameters are needed for our MBS scheme, namely α and σ. These are the size of the masking parameters $\mathbf{a}$ and b which are used to blind the challenge c and un-blind the blinded signature $\mathbf{z}$. Setting parameters for MBS should follows the way done in [DKL+19] which is a trade-off of some factors such as the hardness of the underlying problems MLWE and MSIS, the number of restarts happening locally or globally in the scheme which affects to the runtime, the total size of the public key and the signature and so on. The concrete parameters for MBS in accordance with the security levels as in Dilithium will be considered and specified in the full version of this paper.

VI. Conclusions

We proposed a Dilithium-based blind signature scheme using module lattices in this paper. Basing on MLWE, MSIS hardness assumptions, the proposed scheme is shown to be secure against key recovery attacks as well as to be one-more unforgeable secure in the random oracle model. The blindness of our scheme is obtained using rejection sampling, a typical method used in lattice-based cryptography. The compression technique is also exploited to minimize the public key size and to make the public key+signature size small. A consideration of security of MBS in the QROM model and setting practically concrete parameters for MBS are left to the future works.

Acknowledgment

We would like to thank anonymous reviewers for their helpful comments.

References

[Cha83] David Chaum. Blind Signatures for Untraceable Payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology*, pages 199–203, Boston, MA, 1983. Springer US.

[Cha84] David Chaum. *Blind Signature System*, pages 153–153. Springer US, Boston, MA, 1984.

[Cha88] David Chaum. Blinding for Unanticipated Signatures. In David Chaum and Wyn L. Price, editors, *Advances in Cryptology — EUROCRYPT' 87*, pages 227–233, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

[CPS95] Jan L. Camenisch, Jean-Marc Piveteau, and Markus A. Stadler. Blind signatures based on the discrete logarithm problem. In Alfredo De Santis, editor, *Advances in Cryptology — EUROCRYPT'94*, pages 428–432, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[DKL+19] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium, 2019). Available from: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions. Accessed on April 19, 2019.

[DLL+17] Léo Ducas, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Dilithium: Digital Signatures from Module Lattices. *IACR Cryptology ePrint Archive*, 2017:633, 2017.

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A Concrete Treatment of Fiat-Shamir Signatures in the Quantum Random-Oracle Model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 552–586, Cham, 2018. Springer International Publishing.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, Jun 2015.

[Lyu09] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, pages 598–616, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[NR93] Kaisa Nyberg and Rainer A. Rueppel. A New Signature Scheme Based on the DSA Giving Message Recovery. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, pages 58–61, New York, NY, USA, 1993. ACM.

[Rüc08] Markus Rückert. Lattice-based Blind Signatures. Cryptology ePrint Archive, Report 2008/322, 2008. https://eprint.iacr.org/2008/322.

[Rüc10] Markus Rückert. Lattice-based Blind Signatures. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, pages 413–430, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[Sho97] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997.