BLAZE: Practical Lattice-Based Blind Signatures for Privacy-Preserving Applications

Nabil Alkeilani Alkadri[1], Rachid El Bansarkhani[2], and Johannes Buchmann[1]

[1] Technische Universität Darmstadt, Germany
nabil.alkadri@tu-darmstadt.de, buchmann@cdc.informatik.tu-darmstadt.de
[2] QuantiCor Security GmbH, Germany
rachid.elbansarkhani@quanticor-security.de

Abstract. Blind signatures constitute basic cryptographic ingredients for privacy-preserving applications such as anonymous credentials, e-voting, and Bitcoin. Despite the great variety of cryptographic applications, blind signatures also found their way in real-world scenarios. Due to the expected progress in cryptanalysis using quantum computers, it remains an important research question to find practical and secure alternatives to systems based on classical security assumptions that are not future-proof. In this work we present BLAZE, a new practical blind signature scheme from lattice assumptions. With respect to all relevant efficiency metrics BLAZE is much more efficient than all previous blind signature schemes based on assumptions conjectured to withstand quantum computer attacks. In particular, BLAZE considerably improves upon the first (and currently only secure) lattice-based proposal introduced by Rückert at ASIACRYPT 2010 (RBS). For instance, at 128 bits of security signatures are as small as 6.6 KB, which represents an improvement factor of 13.5 compared to RBS, 2.7 compared to all previous candidates, and an expansion factor of 2.5 compared to the NIST PQC submission Dilithium. We also give a highly optimized implementation, which demonstrates the efficiency of BLAZE to be deployed in practical applications. In particular, generating a blind signature takes just 18 ms, which represents a factor improvement of 15 compared to RBS. The running times for key generation and verification are in the same order as state-of-the-art regular signature schemes, however several orders of magnitudes faster than RBS.

Keywords: Blind Signatures · Lattices · Post-Quantum · Privacy

1 Introduction

Blind signature schemes allow users while interacting with a signer to generate signatures on messages such that the signer gets no information about the message being signed (*blindness*). The user in turn is not able to produce any valid signature without interacting with the signer (*one-more unforgeability*). Blind signatures were proposed by Chaum [Cha82] and have become fundamental building blocks in privacy-oriented cryptography. One of the main applications of blind signatures is anonymous credentials [BL13], which allow users to privately obtain and prove possession of credentials while revealing as little about themselves as possible. This complies with the European privacy standards [PotEU01, PotEU09] and the National Strategy for Trusted Identities in Cyberspace [Coo10]. An established real-life use case of blind signatures in anonymous credentials is the U-Prove technology [Paq13] designed by Microsoft. U-Prove is one of the technologies, to which the Microsoft's Open Specification Promise [Mic07] applies and is integrated for example by Gemalto - a leading digital security company - in its smart card technology in order to enhance privacy [Gem11]. Another application of blind signatures is e-voting systems [KKS17], where authorities blindly sign public keys used by voters to anonymously cast their votes. An e-voting protocol [AG18], which uses blind signatures to verify ballots, is implemented and analyzed on the Ethereum blockchain. Further applications of

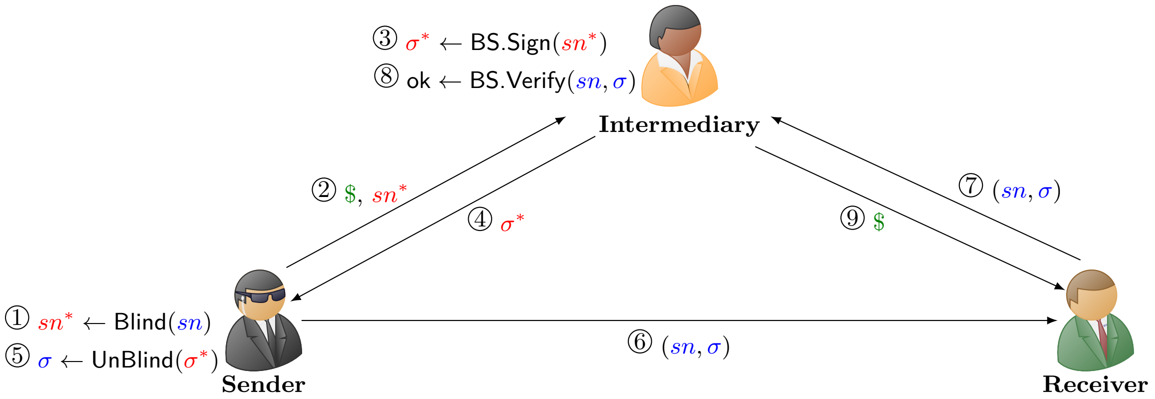

Fig. 1. A simplified protocol for anonymous transactions of digital coins. A sender $\mathcal{S}$ generates a random serial number sn, hides it using an algorithm Blind, and sends the blinded number sn^* together with a coin \$ to a trusted intermediary $\mathcal{I}$, who signs sn^* and sends its signature σ^* back to $\mathcal{S}$. Afterwards, $\mathcal{S}$ applies an algorithm $\mathsf{UnBlind}$ on σ^* to obtain a signature σ on sn, and proceeds by sending the pair (sn, σ) to the receiver $\mathcal{R}$. Later, $\mathcal{R}$ simply forwards (sn, σ) to $\mathcal{I}$, who verifies the validity of the signature and send \$ to $\mathcal{R}$. Privacy is established as $\mathcal{I}$ cannot link the signature σ to $\mathcal{S}$. The algorithms Blind,$\mathsf{UnBlind}$ are realized by any blind signature scheme.

blind signatures include e-cash systems utilizing the Bitcoin blockchain [HBG16], where entities blindly sign digital coins withdrawn by users for selling and buying products and services over the Internet and open networks. Figure 1 illustrates a simplified anonymous payment protocol employing blind signatures.

The above mentioned (real-world) applications rely on classical blind signature schemes, where the security is based on the hardness of number-theoretic assumptions such as RSA and discrete logarithms. For instance, the U-Prove protocol implemented by Gemalto employs blind signature constructions, which are secure as long as computing discrete logarithms is hard [Paq13]. As it is meanwhile known, number-theoretic assumptions are not secure for the long-term, especially when taking into account the recent developments of quantum computers. Consequently, these constructions have to be replaced with blind signature schemes that are comparable in terms of efficiency and secure or at least conjectured to be secure under quantum computer attacks. More concretely, we need post-quantum candidates of blind signature schemes in order to further preserve privacy standards and anonymity considerations. While such proposals do exist [Rüc10, PSM17, BGSS17], they cannot be deployed in practical applications due to their poor performance as well as large keys and signatures (see Table 1). These facts have a significant impact on the efficiency of the applications, especially when implementing blind signatures in constrained devices such as smart cards and wireless sensor networks.

1.1 Our Contributions

In this work we present a new and practical lattice-based blind signature scheme that we call BLAZE. It is based on the Fiat-Shamir with aborts paradigm [Lyu09] and provides statistical blindness and strong one-more unforgeability in the random oracle model (ROM) assuming the hardness of RLWE (ring learning with errors) and RSIS (ring short integer solution) problem. We provide an optimized implementation of BLAZE attesting its practicality and propose parameters targeting 128 and 192 bits of security. Our software implementation and parameters show that BLAZE is much more efficient than the previous blind signature schemes [BGSS17, PSM17, Rüc10] based on assumptions believed to be secure under quantum computer attacks. More precisely, at approximately the same security level BLAZE achieves significant improvement factors with respect to all efficiency metrics including key generation, signing, verification, and sizes of keys and signatures. These factors are shown in

Table 1. Comparison of the existing blind signature schemes that are conjectured to be secure under quantum computer attacks. The table contents are adopted from Section 5, [Rüc10, Table 3], [PSM17, Table 1,2], and [BGSS17, Table 1]. The improvement factor for each efficiency metric, e.g., signature size, is obtained by comparing our scheme BLAZE with the best among the other schemes. We note that only the size of public keys and signatures are given in [BGSS17].

		Sizes in kilo bytes (KB)			Times in milliseconds (ms)		
	Security (bits)	Secret key	Public key	Signature	Key generation	Signing	Verification
This work	128	0.8	3.9	6.6	0.1	17.8	0.1
[Rüc10]	102	**23.6**	23.6	89.4	**52**	**283**	**57**
[PSM17]	102	36.6	54.6	**17.6**	9392	3662	2656
[BGSS17]	100	-	**15**	200	-	-	-
Improvement factor		29.5	3.8	2.7	520	15.9	570

Table 1. The parameters used in our implementation are in the order of current state-of-the-art ordinary signature schemes such as the recent lattice-based NIST submission Dilithium [DKL$^+$18]. In fact, we show in Table 2 that the efficiency of BLAZE is moderately comparable to Dilithium. For instance, a blind signature produced by BLAZE occupies only 6.6 KB of memory, which is larger by a factor of 2.5 compared to Dilithium. This is for example suitable for wireless sensor networks, where it is crucial to decrease the amount of transmitted data in order to reduce the battery power consumption. Furthermore, the fact that BLAZE is *strongly* one-more unforgeable (i.e., the same message may be signed arbitrary many times, which is an important feature for schemes deployed in practice), allows us to prove BLAZE in the new security model *honest-user unforgeability* recently proposed by Schröder and Unruh [SU17, Lemma 10]. It has been shown to be more convenient for blind signature schemes as it removes certain types of attacks not captured in the traditional security model of blind signatures due to Pointcheval and Stern [PS00].

1.2 Our Techniques

In order to give an overview of our techniques, it is instructive to sketch the signing protocol of the blind signature scheme introduced by Rückert [Rüc10] at ASIACRYPT 2010 (RBS), since it is also lattice-based and Fiat-Shamir-like. RBS is one-more unforgeable in the ROM assuming the hardness of RSIS. Its complete description can be found in Appendix A. A signature generated by RBS has the form $(\mathbf{r}, \hat{c}, \hat{z}_1^*, \ldots, \hat{z}_m^*)$ and the signing process works as follows: Upon receiving a “commitment” from the signer $\mathcal{S}$, the user $\mathcal{U}$ hides the signature part $\hat{c}$ output by a random oracle H. Hiding $\hat{c}$ ensures blindness and is accomplished by computing a challenge $\hat{c}^* = \hat{c} - \hat{u}$ for some random secret element $\hat{u}$ and successfully applying rejection sampling to make sure that $\hat{c}^*$ indeed masks $\hat{c}$. Otherwise $\mathcal{U}$ selects a new $\hat{u}$ and repeat until success and proceeds by sending $\hat{c}^*$ to $\mathcal{S}$. Subsequently, $\mathcal{S}$ responds with elements $\hat{z}_1^*, \ldots, \hat{z}_m^*$ after carrying out rejection sampling, which ensures at this point that $\mathcal{S}$’s response does not leak information about the secret key. Then, $\mathcal{U}$ transforms this response into the signature part $(\hat{z}_1, \ldots, \hat{z}_m)$. Here, $\mathcal{U}$ further applies rejection sampling to maintain blindness. More precisely, the polynomials $\hat{z}_i^*$ are concealed within $\hat{z}_i = \hat{z}_i^* - \hat{v}_i$, where $\hat{v}_i$ are uniformly random masking elements chosen by $\mathcal{U}$. Finally, $\mathcal{U}$ sends a signal to $\mathcal{S}$. This signal allows to prove that no valid signature has been obtained in case the last rejection sampling step fails and it further indicates that a protocol restart is required. In addition, the protocol employs statistically hiding and computationally binding commitments to ensure blindness and one-more unforgeability over repetitions. In other words, $\mathcal{U}$ signs a commitment using a randomness $\mathbf{r}$ instead of the message and reveals its opening along with the signature.

The goal of our new design in BLAZE is to improve all relevant sizes and running times as well as security. Our observation is that relying on both RLWE and RSIS (as in state-of-the-art lattice-based

Table 2. Comparing BLAZE and Dilithium [DKL+18] at 128 bits of security. We note that the size of secret keys is not given for Dilithium in [DKL+18].

	Sizes (bytes)			Times (cycles)		
	Secret key	Public key	Signature	Key generation	Signing	Verification
BLAZE	768	3984	6710	$204,671$	$35,547,397$	$276,210$
Dilithium	-	1472	2701	$371,083$	$1,562,215$	$375,708$

schemes) in addition to removing the 1st rejection sampling carried out by $\mathcal{U}$ constitute the main measures towards achieving this goal. The latter is established in BLAZE via a new kind of *partitioning and permutation* technique, which may be of independent interest. It works as follows: Rather than adding the masking term $\hat{u}$ to the challenge $\hat{c}$, we use signed rotation polynomials for masking. The resulting elements still lie in the range of H and are randomized by rotation. Here, it is crucial for H to output elements with exactly κ entries from $\{\pm 1\}$ and $n - \kappa$ entries equal to 0, where n is the number of entries. A random element with entries in other sets may still leak information even after rotation. More formally, let $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and $\hat{p}_j \in R$ $(j = 1, \ldots, \kappa)$ be signed rotation polynomials, i.e., they have the form $\pm x^i$ for some $i \in \mathbb{Z}$. We split the output $\hat{c}$ of H into κ signed rotation polynomials $\hat{c}_1, \ldots, \hat{c}_\kappa$. These polynomials have each a coefficient from $\{\pm 1\}$ and degree at most $n - 1$. Then, we "permute" each part $\hat{c}_j$ using one of the secret polynomials $\hat{p}_j^{-1}$. The resulting elements $\hat{c}_j^*$ will then be signed by $\mathcal{S}$ to $(\hat{z}_{j,1}^*, \hat{z}_{j,2}^*)$. In order for the final signature (output by $\mathcal{U}$) to be successfully verified, we must account for the partitioning and rotating. That is, multiplying the received tuples $(\hat{z}_{j,1}^*, \hat{z}_{j,2}^*)$ each with $\hat{p}_j$ and summing them up with secret masking terms yields the signature part $(\hat{z}_1, \hat{z}_2)$. This technique does not only remove one rejection sampling, it also ensures shorter signatures and speeds up the rejection sampling performed by $\mathcal{S}$. This is because the bound on the norms $\left\|\hat{z}_{j,i}^*\right\|$ becomes significantly smaller. In RBS, the element $\hat{c}^*$ has entries bounded by $n - 1$, whereas BLAZE preserves the norm $\sqrt{\kappa}$ as in state-of-the-art lattice-based signature schemes, e.g., [DDLL13, DKL+18]. Consequently, $\mathcal{S}$ and $\mathcal{U}$ can use smaller masking terms for the remaining two rejection sampling steps and hence the size of the required modulus is also reduced. This already reduces the signature size by a factor of approximately $\log(n)$. We note that κ is much smaller than n and selected such that outputs of H provide enough security.

In case the last rejection sampling fail, we follow RBS and design a proof of failure allowing $\mathcal{U}$ to convince $\mathcal{S}$ that no valid signature has been obtained and hence letting $\mathcal{S}$ restart the protocol. This proof includes all secret elements generated by $\mathcal{U}$ during signing. In order to still ensure statistical blindness, $\mathcal{U}$ signs a commitment τ to the message rather than the message itself and includes its opening in the final signature. The binding property of τ preserves the strong one-more unforgeability.

1.3 Related work

In addition to RBS, there are other lattice-based constructions of blind signatures found in literature. However, we show in Appendix B that they are unfortunately insecure. More precisely, we show for the proposal in [ZTZ+17] how the secret key can simply be recovered already after two executions of its signing protocol. For the rest schemes [CCT+11, ZM14, ZH16, GHWX16, GHW+17] we show that any user is able to solve the underlying lattice problem in just one execution of the signing protocol. Concerning lattice-based constructions, this leaves us with the scheme RBS. Other post-quantum blind signature schemes that we are aware of is the multivariate-based one from [PSM17] and the code-based one proposed in [BGSS17]. Table 1 shows that BLAZE is more efficient than those schemes in terms of all efficiency metrics.

1.4 Outline

In Section 2 we give the background required throughout this work. Then, we present in Section 3 our new blind signature scheme BLAZE. Afterwards, we describe in Section 4 our software implementation of the new scheme. Then, we propose in Section 5 concrete parameters and compare BLAZE with the schemes [BGSS17, PSM17, Rüc10]. Finally, we conclude our results and discuss possible future directions in Section 6.

2 Preliminaries

This section covers the necessary background required throughout this work. First, we give some general notation. Then, we formally define blind signature schemes and their security properties in Section 2.1. Finally, we define lattices and the required lattice problems in Section 2.2.

Notation. We let $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ denote the set of natural numbers, integers, and real numbers, respectively. For a positive integer k, we let $[k]$ denote the set $\{1, 2, \ldots, k\}$. We denote column vectors with bold lower-case letters and matrices with bold upper-case letters. For any positive integer q, we write $\mathbb{Z}_q$ to denote the set of integers in the range $[-\frac{q}{2}, \frac{q}{2}) \cap \mathbb{Z}$. The Euclidean norm ($\ell_2$-norm) of a vector $\mathbf{v}$ with entries v_i is defined as $\|\mathbf{v}\| = (\sum_i |v_i|^2)^{1/2}$, and its ℓ_∞-norm as $\|\mathbf{v}\|_\infty = \max_i |v_i|$. We define the ring $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and its quotient $R_q = R/qR$, where n is power of 2. A ring element $a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} \in R_q$ is denoted by $\hat{a}$ and it corresponds to a vector $\mathbf{a} \in \mathbb{Z}_q^n$ via coefficient embedding. Hence, $\|\hat{a}\| = \|\mathbf{a}\|$ and $\|\hat{a}\|_\infty = \|\mathbf{a}\|_\infty$. We write $\hat{\mathbf{a}} = (\hat{a}_1, \ldots, \hat{a}_k) \in R_q^k$ to denote a vector of ring elements. Its ℓ_2 and ℓ_∞ norm is defined by $\|\hat{\mathbf{a}}\| = (\sum_i^k \|\hat{a}_i\|^2)^{1/2}$ and $\|\hat{\mathbf{a}}\|_\infty = \max_i \|\hat{a}_i\|_\infty$. We let $\mathbb{T}_\kappa^n$ denote the set of all $(n-1)$-degree polynomials with coefficients from $\{-1, 0, 1\}$ and Hamming Weight κ. All logarithms in this work are to base 2, and we always denote the security parameter by $\lambda \in \mathbb{N}$. A function $f : \mathbb{N} \to \mathbb{R}$ is called *negligible* if there exists an $n_0 \in \mathbb{N}$ such that for all $n > n_0$, it holds $f(n) < \frac{1}{p(n)}$ for any polynomial p. With $\text{negl}(\lambda)$ we denote a negligible function in λ. A probability is called overwhelming if it is at least $1 - \text{negl}(\lambda)$. The *statistical distance* between two distributions X, Y over a countable domain D is defined by $\varDelta(X, Y) = \frac{1}{2}\sum_n |X(n) - Y(n)|$. The distributions X, Y are called *statistically close* if $\varDelta(X, Y) = \text{negl}(\lambda)$. We write $x \leftarrow D$ to denote that x is sampled according to a distribution D. By $x \leftarrow_\$ S$ we denote that x is assigned a uniform random element from a finite set S. For two algorithms $\mathcal{A}, \mathcal{B}$ we write $(x, y) \leftarrow \langle \mathcal{A}(a), \mathcal{B}(b)\rangle$ to describe the joint execution of $\mathcal{A}$ and $\mathcal{B}$ in an interactive protocol with private inputs a for $\mathcal{A}$ and b for $\mathcal{B}$ as well as private outputs x for $\mathcal{A}$ and y for $\mathcal{B}$. Accordingly, we write $\mathcal{A}^{\langle \cdot, \mathcal{B}(b)\rangle^k}(a)$ if $\mathcal{A}$ can invoke up to k executions of the protocol with $\mathcal{B}$.

2.1 Blind Signatures and their Security

Definition 1 (Blind Signature Scheme). *A blind signature scheme BS is a tuple of polynomial-time algorithms BS=(BS.KGen,BS.Sign,BS.Verify) such that:*

- *BS.KGen(1^λ) is a key generation algorithm that outputs a pair of keys (pk,sk), where pk is a public (verification) key and sk is a secret (signing) key.*
- *BS.Sign(sk, pk, μ) is an interactive protocol between a signer $\mathcal{S}$ and a user $\mathcal{U}$. The private input of $\mathcal{S}$ is a secret key sk, whereas the private input of $\mathcal{U}$ is a public key pk and a message $\mu \in \mathcal{M}$ with message space $\mathcal{M}$. The private output of $\mathcal{S}$ is a view $\mathcal{V}$ (interpreted as a random variable) and the private output of $\mathcal{U}$ is a signature σ, i.e., $(\mathcal{V}, \sigma) \leftarrow \langle \mathcal{S}(\mathsf{sk}), \mathcal{U}(\mathsf{pk}, \mu)\rangle$. We write $\sigma = \perp$ to denote failure.*

```
**Game** $\mathsf{Blind}_{\mathsf{BS},\mathcal{S}^*}(\lambda)$
$(\mathsf{pk}, \mu_0, \mu_1, \mathsf{state}_{\mathsf{find}}) \leftarrow \mathcal{S}^*(\mathsf{find}, 1^\lambda)$
$b \leftarrow_\$ \{0, 1\}$
$\mathsf{state}_{\mathsf{issue}} \leftarrow \mathcal{S}^{*\langle \cdot, \mathcal{U}(\mathsf{pk}, \mu_b)\rangle^1, \langle \cdot, \mathcal{U}(\mathsf{pk}, \mu_{1-b})\rangle^1}(\mathsf{issue}, \mathsf{state}\ _{\mathsf{find}})$
$\sigma_b := \mathcal{U}(\mathsf{pk}, \mu_b), \sigma_{1-b} := \mathcal{U}(\mathsf{pk}, \mu_{1-b})$
**if** $(\sigma_0 = \perp\ \vee\ \sigma_1 = \perp)$ **then**
    $(\perp, \perp) \leftarrow (\sigma_0, \sigma_1)$
$b^* \leftarrow \mathcal{S}^*(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{state}_{\mathsf{issue}})$
**if** $b^* = b$ **then**
    **return** $1$
**return** $0$

**Game** $\mathsf{Forge}_{\mathsf{BS},\mathcal{U}^*}(\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{BS.KGen}(1^\lambda)$
$\mathsf{H} \leftarrow \mathcal{H}(1^\lambda)$
$((\mu_1, \sigma_1), \cdot\cdot, (\mu_l, \sigma_l)) \leftarrow \mathcal{U}^{*\mathsf{H}(\cdot), \langle \mathcal{S}(\mathsf{sk}), \cdot\rangle^\infty}(\mathsf{pk})$
$k :=$ number of successful signing invocations
**if** $\big(\mu_i \neq \mu_j$ for all $1 \leq i < j \leq l\ \wedge$
        $\mathsf{BS.Verify}(\mathsf{pk}, \mu_i, \sigma_i) = 1, \forall i \in [l]\ \wedge$
        $k + 1 = l\big)$ **then**
    **return** $1$
**return** $0$
```

Fig. 2. Security games of blindness and one-more unforgeability.

- $\mathsf{BS.Verify}(\mathsf{pk}, \mu, \sigma)$ *is a verification algorithm that outputs 1 if the signature* σ *is valid and 0 otherwise.*

Blind signature schemes require the completeness property, i.e., $\mathsf{BS.Verify}$ always (or with overwhelming probability) validates honestly signed messages under honestly created keys. Security of blind signatures is captured by two security notions: blindness and one-more unforgeability [JLO97, PS00]. The former prevents a malicious signer to learn information about user's messages. The latter ensures that each completed execution of $\mathsf{BS.Sign}$ yields at most one signature.

Definition 2 (Blindness). *A blind signature scheme* BS *is called* (t, ε)*-blind if for any adversarial signer* $\mathcal{S}^*$ *running in time at most* t *and working in modes* find*,* issue*, and* guess*, the game* $\mathsf{Blind}_{\mathsf{BS},\mathcal{S}^*}(\lambda)$ *depicted in Figure 2 outputs 1 with probability* $\Pr[\mathsf{Blind}_{\mathsf{BS},\mathcal{S}^*}(\lambda) = 1] \leq \frac{1}{2} + \varepsilon$*, i.e., the advantage of* $\mathcal{S}^*$ *in the game is given by* $\varepsilon = \mathrm{Adv}_{\mathsf{BS},\mathcal{S}^*}(\lambda) = \left|\Pr[b^* = b] - \frac{1}{2}\right|$*. The scheme is statistically blind if it is* $(t = \infty, \varepsilon = \mathrm{negl}(\lambda))$*-blind.*

In the game $\mathsf{Blind}_{\mathsf{BS},\mathcal{S}^*}(\lambda)$, $\mathcal{S}^*$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{BS.KGen}(1^\lambda)$. Then, it chooses two messages μ_0, μ_1 in mode find and sends them along with pk to the honest user $\mathcal{U}$, who randomly chooses a bit b. After that, $\mathsf{BS.Sign}$ is executed twice with $\mathcal{S}^*$ (working in mode issue) and $\mathcal{U}$. Depending on b, $\mathcal{U}$ outputs signatures σ_b, σ_{1-b} in the first and second interaction, respectively. In mode guess, $\mathcal{S}^*$ obtains σ_0, σ_1 in the original order and has to decide which of the two messages has been signed first. We note that this must hold even if $\mathcal{S}^*$ chooses the public key maliciously [ANN06]. If $\mathcal{U}$ outputs $\perp$ in one of both executions, then $\mathcal{S}^*$ is informed about the failure and does not get any signature.

Definition 3 (One-more Unforgeability). *Let* $\mathcal{H}$ *be a family of random oracles. A blind signature scheme* BS *is called* $(t, q_{\mathsf{Sign}}, q_{\mathsf{H}}, \varepsilon)$*-one-more unforgeable in the random oracle model if for any adversarial user* $\mathcal{U}^*$ *running in time at most* t *and making at most* q_{Sign}*,* q_{H} *signing and hash queries, the game* $\mathsf{Forge}_{\mathsf{BS},\mathcal{U}^*}(\lambda)$ *depicted in Figure 2 outputs 1 with probability* $\Pr[\mathsf{Forge}_{\mathsf{BS},\mathcal{U}^*}(\lambda) = 1] \leq \varepsilon$*. The scheme is strongly* $(t, q_{\mathsf{Sign}}, q_{\mathsf{H}}, \varepsilon)$*-one-more unforgeable if the condition* $\mu_i \neq \mu_j$ *in the game changes to* $(\mu_i, \sigma_i) \neq (\mu_j, \sigma_j)$ *for all* $1 \leq i < j \leq l$*.*

In the game $\mathsf{Forge}_{\mathsf{BS},\mathcal{U}^*}(\lambda)$, $\mathcal{U}^*$ tries to output $k + 1$ valid pairs (μ_i, σ_i), for $i \in [k + 1]$, after at most k successful interactions with $\mathcal{S}$.

2.2 Lattices and Gaussians

Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \in \mathbb{R}^{m \times k}$ be a set of linearly independent vectors, where $k \leq m$. The m-dimensional *lattice* $\mathcal{L}$ of rank k generated by $\mathbf{B}$ is given by $\mathcal{L}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^k\} \subset \mathbb{R}^m$. If $m = k$, then $\mathcal{L}$ is *full-rank*. The *determinant* of $\mathcal{L}$, denoted by $\det(\mathcal{L})$, is given by $\sqrt{\det(\mathbf{B}^\top \cdot \mathbf{B})}$, where $\mathbf{B}$ is any basis of $\mathcal{L}$.

The *discrete Gaussian distribution* $D_{\mathcal{L},\sigma,\mathbf{c}}$ over a lattice $\mathcal{L}$ with standard deviation $\sigma > 0$ and center $\mathbf{c} \in \mathbb{R}^n$ is defined as follows: The probability of any $\mathbf{x} \in \mathcal{L}$ is given by $D_{\mathcal{L},\sigma,\mathbf{c}}(\mathbf{x}) = \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\rho_{\sigma,\mathbf{c}}(\mathcal{L})$, where $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(\frac{-\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2})$ and $\rho_{\sigma,\mathbf{c}}(\mathcal{L}) = \sum_{\mathbf{x}\in\mathcal{L}} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. The subscript $\mathbf{c}$ is taken to be $\mathbf{0}$ when omitted. The following two lemmas are central results used throughout this work. The first one gives a tail bound on Gaussian distributed elements, while the second one concerns rejection sampling.

Lemma 1 ([Lyu12, Lemma 4.4]). *For any $t, \eta > 0$ we have*

1. $\Pr_{x \leftarrow D_{\mathbb{Z},\sigma}}[|x| > t \cdot \sigma] \leq 2\exp(-t^2/2)$.
2. $\Pr_{\mathbf{x} \leftarrow D_{\mathbb{Z}^m,\sigma}}[\|\mathbf{x}\| > \eta\sigma\sqrt{m}] \leq \eta^m \exp(\frac{m}{2}(1-\eta^2))$.

Lemma 2 ([Lyu12, Theorem 4.6, Lemma 4.7]). *Let $V \subseteq \mathbb{Z}^m$ with elements having norms bounded by T, $\sigma = \omega(T\sqrt{\log m})$, and $h : V \to \mathbb{R}$ be a probability distribution. Then there exits a constant $M = O(1)$ such that*
$\forall \mathbf{v} \in V : \Pr[D_{\mathbb{Z}^m,\sigma}(\mathbf{z}) \leq M \cdot D_{\mathbb{Z}^m,\sigma,\mathbf{v}}(\mathbf{z});\ \mathbf{z} \leftarrow D_{\mathbb{Z}^m,\sigma}] \geq 1 - \varepsilon$, where $\varepsilon = 2^{-\omega(\log m)}$. Furthermore, the following two algorithms are within statistical distance $\delta = \varepsilon/M$.

1. *$\mathbf{v} \leftarrow h$, $\mathbf{z} \leftarrow D_{\mathbb{Z}^m,\sigma,\mathbf{v}}$, output $(\mathbf{z}, \mathbf{v})$ with probability $\frac{D_{\mathbb{Z}^m,\sigma}(\mathbf{z})}{M \cdot D_{\mathbb{Z}^m,\sigma,\mathbf{v}}(\mathbf{z})}$.*
2. *$\mathbf{v} \leftarrow h$, $\mathbf{z} \leftarrow D_{\mathbb{Z}^m,\sigma}$, output $(\mathbf{z}, \mathbf{v})$ with probability $1/M$.*

Moreover, the probability that the first algorithm outputs something is at least $(1-\varepsilon)/M$. If $\sigma = \alpha T$ for any positive α, then $M = \exp(\frac{12}{\alpha} + \frac{1}{2\alpha^2})$ with $\varepsilon = 2^{-100}$.

We let $\mathsf{RejSamp}(x)$ denote an algorithm that carries out rejection sampling on input x. It outputs 1 if it accepts and 0 otherwise. We write $\mathsf{RejSamp}(x; r)$ to specify the randomness r used within the algorithm. Next, we define the related lattice problems.

Definition 4 (Ring Short Integer Solution (RSIS) Problem). *Let n, q, k be positive integers and β a positive real. Given a uniformly random vector $\hat{\mathbf{a}} = (\hat{a}_1, \ldots, \hat{a}_k) \in R_q^k$, the Hermite Normal Form of RSIS problem asks to find a non-zero vector $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_{k+1}) \in R^{k+1}$ such that $[\hat{\mathbf{a}}\ \mathbf{1}] \cdot \hat{\mathbf{x}} = 0 \pmod q$, where $\|\hat{\mathbf{x}}\| \leq \beta$. The inhomogeneous RSIS asks to find $\hat{\mathbf{x}} \in R^{k+1}$ with $\|\hat{\mathbf{x}}\| \leq \beta$ such that $[\hat{\mathbf{a}}\ \mathbf{1}] \cdot \hat{\mathbf{x}} = \hat{u} \pmod q$, for a given $\hat{u} \in R_q$.*

Definition 5 (Ring Learning With Errors (RLWE) Problem). *Given $\mathrm{poly}(n)$ samples $(\hat{a}_i, \hat{b}_i) \in R_q \times R_q$, the decision RLWE problem asks to distinguish, with non-negligible advantage, whether $(\hat{a}_i, \hat{b}_i)$ were chosen from the uniform distribution over $R_q \times R_q$ or from the distribution that outputs $(\hat{a}, \hat{b} = \langle \hat{a}, \hat{s} \rangle + \hat{e} \pmod q)$ for $\hat{a} \leftarrow_\$ R_q, \hat{s} \leftarrow_\$ R_q$, and $\hat{e} \leftarrow \chi$, where χ is an error distribution over R. The secret $\hat{s}$ can also be chosen from χ. The search RLWE problem asks to find $\hat{s}$.*

Any instance I of the above defined problems is called (t, ε)-hard if any algorithm $\mathcal{A}$ running in time at most t can solve I with probability ε.

3 BLAZE: The New Blind Signature Scheme

In this section we present BLAZE: our new and practical blind signature scheme. It is statistically blind and its strong one-more unforgeability is based on the hardness of RLWE and RSIS problem in the ROM. As opposed to RBS, BLAZE has to pass 2 rejection sampling procedures rather than 3; one is performed by the signer to conceal the secret key and one by the user to achieve blindness. That is, we remove one rejection sampling step from the user side by splitting the challenge generated by the user into monomials with entries from $\{-1, 1\}$ and permuting them using secret monomials with entries from $\{-1, 1\}$ as well.

We first introduce new tools and technical lemmas employed within BLAZE.

Definition 6. *Define by $\hat{\mathbb{T}} = \{(-1)^s \cdot x^i \mid \text{for } s \in \mathbb{N} \text{ and } i \in \mathbb{Z}\}$ the set of signed permutation polynomials which represent a rotation multiplied by a sign.*

Lemma 3. *Let $\hat{p} \in \hat{\mathbb{T}}$ with $\hat{p} = (-1)^s \cdot x^i$ for some $i \in \mathbb{Z}$ and $s \in \{0, 1\}$. Then, $\hat{\mathbb{T}}$ is a group with respect to multiplication and the inverse of $\hat{p}$ is given by $\hat{p}^{-1} = (-1)^{1-s} \cdot x^{n-i} \in \hat{\mathbb{T}}$.*

Proof. Let $\hat{p}_1 = (-1)^{s_1} \cdot x^{i_1}$, $\hat{p}_2 = (-1)^{s_2} \cdot x^{i_2} \in \hat{\mathbb{T}}$, then $\hat{p}_1 \cdot \hat{p}_2 = (-1)^{s_1+s_2} \cdot x^{i_1+i_2} \in \hat{\mathbb{T}}$. A simple calculation shows that $\hat{p} \cdot \hat{p}^{-1} = (-1)^s \cdot x^i \cdot (-1)^{1-s} \cdot x^{n-i} = -x^n \equiv 1 \bmod \langle x^n + 1 \rangle$. Thus, every $\hat{p} \in \hat{\mathbb{T}}$ has an inverse $\hat{p}^{-1} \in \hat{\mathbb{T}}$ and the neutral element is given by the constant polynomial 1. □

The following lemma helps proving that partitions of a challenge $\hat{c}$ multiplied with a signed rotation are independent from the initial challenge. This will be used when proving blindness.

Lemma 4. *Let $\hat{c} \in \mathbb{T}^n_\kappa$ and $\hat{c}_1, \ldots, \hat{c}_\kappa$ be a partition such that $\hat{c} = \sum_1^\kappa \hat{c}_i$ and each $\hat{c}_i$ contains exactly the i-th non-zero entry of $\hat{c}$ at exactly the same position. Furthermore, let $\hat{c}_1^* = \hat{p}_1^{-1} \cdot \hat{c}_1, \ldots, \hat{c}_\kappa^* = \hat{p}_\kappa^{-1} \cdot \hat{c}_\kappa$ for random signed rotations $\hat{p}_1, \ldots, \hat{p}_\kappa \in \hat{\mathbb{T}}$. Then, $\hat{c}_i^*, \hat{c}_i \in \hat{\mathbb{T}}$ and for any $\hat{d} \in \mathbb{T}^n_\kappa$ we have*

$$\Pr_{\hat{p}_i \leftarrow_\$ \hat{\mathbb{T}}}[(\hat{c}_1^*, \ldots, \hat{c}_\kappa^*) = (\hat{p}_1^{-1}\hat{c}_1, \ldots, \hat{p}_\kappa^{-1}\hat{c}_\kappa) \mid \hat{c}] = \tag{1a}$$

$$\Pr_{\hat{p}_i, \hat{c}_i \leftarrow_\$ \hat{\mathbb{T}}}[(\hat{c}_1^*, \ldots, \hat{c}_\kappa^*) = (\hat{p}_1^{-1}\hat{c}_1, \ldots, \hat{p}_\kappa^{-1}\hat{c}_\kappa)] \quad = (2n)^{-\kappa} \tag{1b}$$

Proof. For any partitioning we have $\hat{c}_i \in \hat{\mathbb{T}}$, since it contains only one ±1 at exactly the same position as $\hat{c}$. Furthermore, $\hat{c}_i \in \hat{\mathbb{T}}$ can be transformed into any element of $\hat{\mathbb{T}}$ via a signed rotation $\hat{p} \in \hat{\mathbb{T}}$. Let $\hat{c}$ be any element from $\mathbb{T}^n_\kappa$ and $\hat{c}_1, \ldots, \hat{c}_\kappa$ be any partition of $\hat{c}$. Then, for any fixed $\hat{c}_i^* \in \hat{\mathbb{T}}$ there exists exactly one set of elements $\hat{p}_1^{-1}, \ldots, \hat{p}_\kappa^{-1} \in \hat{\mathbb{T}}$ such that $\hat{c}_1^* = \hat{p}^{-1}\hat{c}_1, \ldots, \hat{c}_\kappa^* = \hat{p}_\kappa^{-1}\hat{c}_\kappa$. Thus, probability (1a) evaluates to $(2n)^{-\kappa}$. Next, we recall that for any fixed $\hat{c}_i^* \in \hat{\mathbb{T}}$ and fixed $\hat{c}_i \in \hat{\mathbb{T}}$ there exists exactly one $\hat{p}_i \in \hat{\mathbb{T}}$ such that $\hat{c}_i^* = \hat{p}_i^{-1}\hat{c}_i$. Thus, probability (1b) evaluates to

$$\sum_{\hat{c} \in \mathbb{T}^n_\kappa} \Pr_{\hat{p}_i \leftarrow_\$ \hat{\mathbb{T}}}[(\hat{c}_1^*, \ldots, \hat{c}_\kappa^*) = (\hat{p}_1^{-1}\hat{c}_1, \ldots, \hat{p}_\kappa^{-1}\hat{c}_\kappa) \mid \hat{c}] \cdot P[\hat{c}] = (2n)^{-\kappa} \ .$$

□

In the following we give a detailed description of our new blind signature scheme BLAZE. We let Expand be a public random function on λ-bit strings (e.g., a pseudorandom function). It takes a random input seed and expands it to any desired length. This function is solely used for saving bandwidth as it is deterministic, i.e., given an input it always produces the same output. We let H be a public hash function

```
$\boxed{\mathsf{BS.KGen}(1^\lambda)}$
seed $\leftarrow_\$ \{0,1\}^\lambda$
$\hat{a} \in R_q \leftarrow$ Expand(seed)
$\hat{s}_1, \hat{s}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$
$\hat{b} \leftarrow \hat{a}\hat{s}_1 + \hat{s}_2 \pmod q$
sk $:= (\hat{s}_1, \hat{s}_2)$, pk $:=$ (seed, $\hat{b}$)
return (sk, pk)
```

```
$\boxed{\mathsf{BS.Verify}(\mathsf{seed}, \hat{b}, \mu, (\tau', \mathbf{r}, \hat{z}_1, \hat{z}_2, \hat{c}))}$
$\hat{a} \leftarrow$ Expand(seed)
$(\hat{z}_1, \hat{z}_2) \leftarrow$ Decompress$(\hat{z}_1, \hat{z}_2)$
if $\big(\|(\hat{z}_1, \hat{z}_2)\| \le B \ \wedge\ \hat{c} =$
        $\mathsf{H}(\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c} \pmod q), \tau', \mathsf{Com}(\mu; \mathbf{r}))\big)$
    return 1
return 0
```

```
$\boxed{\mathsf{BS.Sign}(\hat{s}_1, \hat{s}_2, \mathsf{seed}, \mu)}$
Signer $\mathcal{S}(\hat{s}_1, \hat{s}_2)$                                   User $\mathcal{U}(\mathsf{seed}, \mu)$
$\hat{a} \leftarrow$ Expand(seed)
$\hat{y}^*_{1,1}, \ldots, \hat{y}^*_{\kappa,1} \leftarrow D^n_{\mathbb{Z},s^*}$
$\hat{y}^*_{1,2}, \ldots, \hat{y}^*_{\kappa,2} \leftarrow D^n_{\mathbb{Z},s^*}$
for $j = 1, \ldots, \kappa$ :
    $\hat{y}_j \leftarrow \hat{a}\hat{y}^*_{j,1} + \hat{y}^*_{j,2} \pmod q$
$\xrightarrow{\hat{y}_1, \ldots, \hat{y}_\kappa}$
                                                    $\mathbf{r}, \mathbf{r}', \rho \leftarrow_\$ \{0,1\}^\lambda$, $\hat{a} \leftarrow$ Expand(seed)
                                                    $\tau \leftarrow \mathsf{Com}(\mu; \mathbf{r})$, $\tau' \leftarrow \mathsf{Com}(\rho; \mathbf{r}')$
                                                    $\hat{p}_1, \ldots, \hat{p}_\kappa \leftarrow_\$ \hat{\mathbb{T}}$, $\hat{e}_1, \hat{e}_2 \leftarrow D^n_{\mathbb{Z},s}$
                                                    $\hat{c} \leftarrow \mathsf{H}(\hat{a}\hat{e}_1 + \hat{e}_2 + \sum_1^\kappa \hat{p}_j\hat{y}_j \pmod q), \tau', \tau)$
                                                    $\hat{c} := \sum_1^\kappa \hat{c}_j,\ \hat{c}_j \in \hat{\mathbb{T}}$
                                                    for $j = 1, \ldots, \kappa$ : $\hat{c}^*_j \leftarrow \hat{p}_j^{-1} \cdot \hat{c}_j$
                                                    $\xleftarrow{\hat{c}^*_1, \ldots, \hat{c}^*_\kappa}$
for $j = 1, \ldots, \kappa$ :
    $\hat{z}^*_{j,1} \leftarrow \hat{y}^*_{j,1} + \hat{s}_1\hat{c}^*_j$, $\hat{z}^*_{j,2} \leftarrow \hat{y}^*_{j,2} + \hat{s}_2\hat{c}^*_j$
if RejSamp$(\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2}) = 0$ restart
$\xrightarrow{\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2}}$
                                                    $\hat{v}_1 \leftarrow \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,1}$, $\hat{v}_2 \leftarrow \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,2}$
                                                    $\hat{z}_1 \leftarrow \hat{e}_1 + \hat{v}_1$, $\hat{z}_2 \leftarrow \hat{e}_2 + \hat{v}_2$
                                                    if RejSamp$(\hat{z}_1, \hat{z}_2; \rho) = 1$
                                                        result $\leftarrow$ ok
                                                        $(\hat{z}_1, \hat{z}_2) \leftarrow$ Compress$(\hat{z}_1, \hat{z}_2)$
                                                    else result $\leftarrow (\tau, \rho, \mathbf{r}', \hat{p}_1, \ldots, \hat{p}_\kappa, \hat{e}_1, \hat{e}_2, \hat{c})$
                                                    $\xleftarrow{\text{result}}$
if result $\neq$ ok
    $\tau' \leftarrow \mathsf{Com}(\rho; \mathbf{r}')$, $\hat{z}_1 \leftarrow \hat{e}_1 + \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,1}$, $\hat{z}_2 \leftarrow \hat{e}_2 + \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,2}$
    if $\big(\sum_1^\kappa \hat{p}_j\hat{c}^*_j = \hat{c} = \mathsf{H}(\hat{a}\hat{e}_1 + \hat{e}_2 + \sum_1^\kappa \hat{p}_j\hat{y}_j \pmod q), \tau', \tau) \wedge$
        $\hat{c} = \mathsf{H}(\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c} \pmod q), \tau', \tau) \wedge$
        RejSamp$(\hat{z}_1, \hat{z}_2; \rho) = 0\big)$ restart
                                                    return $(\mu, (\tau', \mathbf{r}, \hat{z}_1, \hat{z}_2, \hat{c}))$
```

Fig. 3. A description of the new blind signature scheme BLAZE.

modeled as a random oracle and randomly chosen from the family $\{\mathsf{H} : \{0,1\}^* \to \mathbb{T}^n_\kappa\}$. We further let Com : $\{0,1\}^* \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a statistically hiding and computationally binding commitment function. Finally, we let Compress and Decompress be functions for (de)compressing Gaussian elements (see Section 4 for description). The respective algorithms of BLAZE are formalized in Figure 3.

Key Generation. Given 1^λ the algorithm chooses a uniform random seed $\in \{0,1\}^\lambda$ and expands it to a polynomial $\hat{a} \in R_q$ using Expand. The secret key consists of two polynomials sk $= (\hat{s}_1, \hat{s}_2)$ chosen from $D^n_{\mathbb{Z},\sigma}$, while the public key is given by pk $=$ (seed, $\hat{b} = \hat{a}\hat{s}_1 + \hat{s}_2 \pmod q$).

Signing. Given sk, seed, and a message μ the signer $\mathcal{S}$ samples 2κ masking terms $\hat{y}^*_{j,1}, \hat{y}^*_{j,2} \leftarrow D^n_{\mathbb{Z},s^*}$ for $j \in [\kappa]$ and sends $\hat{y}_j = \hat{a}\hat{y}^*_{j,1} + \hat{y}^*_{j,2} \pmod q$ to the user $\mathcal{U}$. Upon receiving the commitments $\hat{y}_1, \ldots, \hat{y}_\kappa$, $\mathcal{U}$ computes $\tau = \mathsf{Com}(\mu; \mathbf{r})$ and $\tau' = \mathsf{Com}(\rho; \mathbf{r}')$ for random $\mathbf{r}, \mathbf{r}', \rho \in \{0,1\}^\lambda$, $\hat{a} = \mathsf{Expand}(\mathsf{seed})$, and selects random elements $\hat{p}_1, \ldots, \hat{p}_\kappa \in \hat{\mathbb{T}}$ and polynomials $\hat{e}_1, \hat{e}_2$ from $D^n_{\mathbb{Z},s}$. Then, $\mathcal{U}$ generates $\hat{c} = \mathsf{H}(\hat{a}\hat{e}_1 + \hat{e}_2 + \sum_1^\kappa \hat{p}_i\hat{y}_i \pmod q, \tau', \tau) \in \mathbb{T}^n_\kappa$. Subsequently, $\mathcal{U}$ splits $\hat{c}$ into partitions $\hat{c}_1, \ldots, \hat{c}_\kappa \in \hat{\mathbb{T}}$ such that $\hat{c} = \sum_1^\kappa \hat{c}_j$ and the j^{th} partition $\hat{c}_j$ contains the j^{th} non-zero entry of $\hat{c}$ at exactly the same position. Then, $\mathcal{U}$ masks each partition $\hat{c}_j$ by computing $\hat{c}^*_j = \hat{p}_j^{-1} \cdot \hat{c}_j$ for all $j \in [\kappa]$ using the signed rotations $\hat{p}_1, \ldots, \hat{p}_\kappa$. Upon receiving the partitions $\hat{c}^*_j$, $\mathcal{S}$ signs them. To this end, $\mathcal{S}$ computes $\hat{z}^*_{j,1} = \hat{y}^*_{j,1} + \hat{s}_1\hat{c}^*_j$ and $\hat{z}^*_{j,2} = \hat{y}^*_{j,2} + \hat{s}_2\hat{c}^*_j$. Subsequently, $\mathcal{S}$ proceeds by applying rejection sampling ($\mathsf{RejSamp}$) and making sure that $\hat{z}^*_{j,1}, \hat{z}^*_{j,2}$ leak no information about sk. If $\mathsf{RejSamp}$ outputs 0, $\mathcal{S}$ restarts the protocol. Upon receiving $\hat{z}^*_{j,1}, \hat{z}^*_{j,2}$ for $j \in [\kappa]$, $\mathcal{U}$ computes $\hat{v}_1 = \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,1}$ and $\hat{v}_2 = \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,2}$. In order for the verification to succeed, the signature part $(\hat{z}_1, \hat{z}_2)$ output by $\mathcal{U}$ must be brought into the form $\hat{z}_1 = \hat{y}^*_1 + \hat{s}_1\hat{c}$, $\hat{z}_2 = \hat{y}^*_2 + \hat{s}_2\hat{c}$ for some polynomials $\hat{y}^*_1, \hat{y}^*_2$. This is attained by multiplying $\hat{z}^*_{j,1}, \hat{z}^*_{j,2}$ with the elements $\hat{p}_j$, summing them up with the masking terms $\hat{e}_1, \hat{e}_2$, and apply $\mathsf{RejSamp}$ to conceal the distribution of $\hat{z}^*_{j,1}, \hat{z}^*_{j,2}$ from $\mathcal{S}$. Thus, $\mathcal{U}$ must already have taken this into account via the inputs to H after the first move. In fact, we must have $\hat{a}\hat{y}^*_1 + \hat{y}^*_2 = \hat{a}\hat{e}_1 + \hat{e}_2 + \sum_1^\kappa \hat{p}_j\hat{y}_j \pmod q$. Therefore, $\mathcal{U}$ sets $\hat{z}_1 = \hat{e}_1 + \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,1}$ and $\hat{z}_2 = \hat{e}_2 + \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,2}$. Finally, $\mathcal{U}$ compresses $(\hat{z}_1, \hat{z}_2)$ using $\mathsf{Compress}$ and sends $\mathsf{result} = \mathsf{ok}$ to $\mathcal{S}$. The signature is given by $(\tau', \mathbf{r}, \hat{z}_1, \hat{z}_2, \hat{c})$. If $\mathsf{RejSamp}$ outputs 0, $\mathcal{U}$ sends $\mathcal{S}$ a proof of failure by setting $\mathsf{result} = (\tau, \rho, \mathbf{r}', \hat{p}_1, \ldots, \hat{p}_\kappa, \hat{e}_1, \hat{e}_2, \hat{c})$. This allows $\mathcal{S}$ to perform 3 checks (see Figure 3) in order to verify that $\mathcal{U}$ has not obtained a valid signature and hence restarts the protocol. Note that the randomness ρ used in the last rejection sampling must be part of the proof of failure. However, it cannot be part of the signature, since it may leak information about the secret terms involved in computing $\hat{z}_1, \hat{z}_2$. This is why BLAZE includes the commitment τ' in the signature as well as in the input of H generating $\hat{c}$ in order to preserve security.

Verification. On input $(\mathsf{seed}, \hat{b}, \mu, (\tau', \mathbf{r}, \hat{z}_1, \hat{z}_2, \hat{c}))$ the verifier uses Expand to compute $\hat{a}$ out of seed, decompresses $(\hat{z}_1, \hat{z}_2)$ using $\mathsf{Decompress}$. It accepts if and only if $\|(\hat{z}_1, \hat{z}_2)\|$ is smaller than some predefined bound B and the output of H on $\left(\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c} \pmod q, \tau', \mathsf{Com}(\mu; \mathbf{r})\right)$ is equal to $\hat{c}$.

In the following we prove completeness, blindness, and strong one-more unforgeability of BLAZE.

Theorem 1. *Let Com be a statistically hiding and computationally binding commitment function. Let $\alpha^*, \alpha, \eta > 0$, $s^* = \alpha^*\sqrt{\kappa} \cdot \|(\hat{s}_1, \hat{s}_2)\|$, $s = \eta\alpha\sqrt{2\kappa n}s^*$, and $B = \eta s\sqrt{2n}$. After at most $M = M_{\mathcal{S}} \cdot M_{\mathcal{U}}$ repetitions, any blind signature produced by BLAZE is validated with probability at least $1 - 2^{-\lambda}$, where $M_{\mathcal{S}} = \exp(\frac{12}{\alpha^*} + \frac{1}{2\alpha^{*2}})$ and $M_{\mathcal{U}} = \exp(\frac{12}{\alpha} + \frac{1}{2\alpha^2})$ are the expected number of repetitions by the signer and user, respectively.*

Proof. For an honestly generated signature $(\tau', \mathbf{r}, \hat{z}_1, \hat{z}_2, \hat{c})$, the pair $(\hat{z}_1, \hat{z}_2)$ is distributed according to $D^{2n}_{\mathbb{Z},s}$ and bounded by $\eta s\sqrt{2n} = B$ with probability $1 - \eta^{2n}\exp(n(1-\eta^2))$ (Lemma 1). By choosing η such that this probability $\leq 2^{-\lambda}$ we have $\|(\hat{z}_1, \hat{z}_2)\| \leq B$ with probability $1 - 2^{-\lambda}$. The condition $\mathsf{H}(\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c} \pmod q, \tau', \tau) = \hat{c}$ is satisfied due to the correctness of Com and the following:

$$\begin{aligned}
\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c} &= \hat{a}\Big(\hat{e}_1 + \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,1}\Big) + \Big(\hat{e}_2 + \sum_1^\kappa \hat{p}_j\hat{z}^*_{j,2}\Big) - \hat{b}\hat{c} \\
&= \hat{a}\Big(\hat{e}_1 + \sum_1^\kappa (\hat{s}_1\hat{c}_j + \hat{p}_j\hat{y}^*_{j,1})\Big) + \hat{e}_2 + \sum_1^\kappa (\hat{s}_2\hat{c}_j + \hat{p}_j\hat{y}^*_{j,2}) - \hat{b}\hat{c} \\
&= \hat{a}\hat{e}_1 + \hat{e}_2 + \sum_1^\kappa \hat{p}_j\left(\hat{a}\hat{y}^*_{j,1} + \hat{y}^*_{j,2}\right) + \hat{c}\left(\hat{a}\hat{s}_1 + \hat{s}_2\right) - \hat{b}\hat{c} \\
&= \hat{a}\hat{e}_1 + \hat{e}_2 + \sum_1^\kappa \hat{p}_j\hat{y}_j \pmod q \ .
\end{aligned}$$

Next, applying rejection sampling (Lemma 2) by the signer accepts with probability

$$D_{\mathbb{Z}^{2\kappa n},s^*}(\mathbf{z}^*)/(M_{\mathcal{S}} D_{\mathbb{Z}^{2\kappa n},s^*,\mathbf{v}^*}(\mathbf{z}^*)),$$

where $\mathbf{z}^*, \mathbf{v}^*$ are the vector representations of $(\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2})$, $(\hat{s}_1\hat{c}^*_1, \ldots, \hat{s}_1\hat{c}^*_\kappa, \hat{s}_2\hat{c}^*_1, \ldots, \hat{s}_2\hat{c}^*_\kappa)$ and the expected number of repetitions is given by $M_{\mathcal{S}} = \exp(\frac{12}{\alpha^*} + \frac{1}{2\alpha^{*2}})$ for $s^* = \alpha^* \|\mathbf{v}^*\| = \alpha^* \sqrt{\kappa} \|(\hat{s}_1, \hat{s}_2)\|$. Finally, rejection sampling performed by the user side accepts with probability $D_{\mathbb{Z}^{2n},s}(\mathbf{z})/(M_{\mathcal{U}} D_{\mathbb{Z}^{2n},s,\mathbf{v}}(\mathbf{z}))$, where $\mathbf{z}, \mathbf{v}$ are the vector representations of $(\hat{z}_1, \hat{z}_2)$, $(\sum_1^\kappa \hat{p}_j \hat{z}^*_{j,1}, \sum_1^\kappa \hat{p}_j \hat{z}^*_{j,2})$ and the expected number of repetitions is $M_{\mathcal{U}} = \exp(\frac{12}{\alpha} + \frac{1}{2\alpha^2})$ for $s = \alpha \|\mathbf{v}\|$. The entries of $\mathbf{v}$ are distributed according to $D^n_{\mathbb{Z},\sqrt{\kappa}s^*}$ (see [BF11, Theorem 9]). Hence, $\|\mathbf{v}\| \leq \eta\sqrt{2\kappa n}s^*$ and $s = \eta\alpha\sqrt{2\kappa n}s^*$. Therefore, the total expected number of repetitions is $M = M_{\mathcal{S}} \cdot M_{\mathcal{U}}$. □

Theorem 2. *Let Com be a statistically hiding and computationally binding commitment function. The scheme BLAZE is $(t = \infty, \varepsilon = \frac{2^{-100}}{M_{\mathcal{U}}})$-blind.*

Proof. In the game $\mathsf{Blind}_{\mathsf{BS},\mathcal{S}^*}(\lambda)$ given in Definition 2 the adversarial signer $\mathcal{S}^*$ selects two messages μ_0, μ_1 and interacts with the user $\mathcal{U}$ twice, i.e., $\mathcal{U}(\mathsf{seed}, \mu_b)$ in the first run and subsequently $\mathcal{U}(\mathsf{seed}, \mu_{1-b})$ for a random bit b chosen by $\mathcal{U}$. We show that after each interaction, $\mathcal{U}$ does not leak any information about the respective message being signed. More precisely, the exchanged messages during protocol execution together with the user's output (interpreted as random variables) are independently distributed, especially also from the message being signed. This requires analyzing only the signature part $(\hat{z}_1, \hat{z}_2)$, since τ' is a statistically hiding commitment, $\mathbf{r}$ is uniformly random, $\hat{c} \in \mathbb{T}^n_\kappa$ and $\hat{c}^*_1, \ldots, \hat{c}^*_\kappa \in \hat{\mathbb{T}}$ are uniformly random and independently distributed by Lemma 4.

Let $(\hat{z}_1, \hat{z}_2)_b$ and $(\hat{z}_1, \hat{z}_2)_{1-b}$ be the signature parts output by $\mathcal{U}(\mathsf{seed}, \mu_b)$ and $\mathcal{U}(\mathsf{seed}, \mu_{1-b})$, respectively. They have the form $(\hat{z}_1, \hat{z}_2) = (\hat{e}_1 + \sum_1^\kappa \hat{p}_j \hat{z}^*_{j,1}, \hat{e}_2 + \sum_1^\kappa \hat{p}_j \hat{z}^*_{j,2})$, where $\hat{p}_1, \ldots, \hat{p}_\kappa$ are uniform random elements from $\hat{\mathbb{T}}$, the polynomials $\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2}$ are each distributed as $D^n_{\mathbb{Z},s^*}$, and $\hat{e}_1, \hat{e}_2$ are distributed according to $D_{\mathbb{Z}^n,s}$. When applying rejection sampling (Lemma 2) on the pairs $(\hat{z}_1, \hat{z}_2)_b, (\hat{z}_1, \hat{z}_2)_{1-b}$, they completely hide $(\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2})_b, (\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2})_{1-b}$, respectively, and become independently distributed within statistical distance of $\frac{2^{-100}}{M_{\mathcal{U}}}$ from $D^{2n}_{\mathbb{Z},s}$. Finally, we note that if the protocol needs to be restarted, then the user selects fresh $\mathbf{r}, \mathbf{r}', \rho, \hat{p}_1, \ldots, \hat{p}_\kappa$, and $\hat{e}_1, \hat{e}_2$. Therefore, protocol executions are independent of each other and hence the signer does not get information about the message being signed. Moreover, the proof of failure also maintains blindness due to the statistical hiding property of Com. □

Remark 1. Similar to RBS, we note that BLAZE remains blind under the stronger blindness definition given in [ANN06], i.e., even if pk is chosen maliciously by $\mathcal{S}^*$. This is because the above proof does not exploit any special features of the key. Furthermore, selective failure blindness [CNS07] is already achieved since a commitment to the message is being signed using a statistically hiding commitment scheme [FS09].

Recovering the secret key of BLAZE is as hard as RLWE. Thus, we prove its strong one-more unforgeability assuming the hardness of RLWE, i.e., we assume that the public key $(\hat{a}, \hat{b})$ is chosen uniformly at random.

Theorem 3. *Let Com be a statistically hiding and computationally binding commitment function. The scheme BLAZE is strongly $(t_{\mathcal{A}}, q_{\mathsf{Sign}}, q_H, \varepsilon_{\mathcal{A}})$-one-more unforgeable if (inhomogeneous) RSIS is $(t_{\mathcal{D}}, \varepsilon_{\mathcal{D}})$-hard. That is, if it is hard to find $(\hat{v}_1, \hat{v}_2, \hat{v}_3) \neq 0$ such that $\|(\hat{v}_1, \hat{v}_2)\| \leq 2B + s/\alpha$ and $\|\hat{v}_3\|_\infty \leq 2$ satisfying $\hat{a}\hat{v}_1 + \hat{v}_2 = \hat{v}_3\hat{b} \pmod q$, where $t_{\mathcal{D}} \leq t_{\mathcal{A}} + q_H^{q_{\mathsf{Sign}}}(q_{\mathsf{Sign}} + q_H)$, $\varepsilon_{\mathcal{D}} \geq \min\{\frac{\varepsilon_{\mathsf{fork}}}{2(k+1)}, \varepsilon_{\mathsf{abort}}\}$, and $k \leq q_{\mathsf{Sign}}$ denotes the successful signing queries. The probabilities $\varepsilon_{\mathsf{fork}}, \varepsilon_{\mathsf{abort}}$ are given in the proof. The signing algorithm produces a signature with probability $(1 - 2^{-100})/M$, where M is the average repetition rate of the signing protocol.*

Proof. We assume that there exists a forger $\mathcal{A}$ that wins the one-more unforgeability game given in Definition 3 with probability $\varepsilon_{\mathcal{A}}$. We construct a reduction algorithm $\mathcal{D}$ that finds $(\hat{v}_1, \hat{v}_2, \hat{v}_3) \neq 0$ as described in the theorem statement with probability $\varepsilon_{\mathcal{D}}$.

Setup. The input of $\mathcal{D}$ is a random pair $(\hat{a}, \hat{b}) \in R_q \times R_q$. The reduction $\mathcal{D}$ then randomly selects answers for random oracle queries $\{\hat{c}_1, \dots, \hat{c}_{q_{\mathsf{H}}}\}$. Then, it runs the forger $\mathcal{A}$ with input $(\hat{a}, \hat{b})$.

Random Oracle Query. The reduction $\mathcal{D}$ maintains a list L_{H}, which includes pairs of random oracle queries and their answers from $\mathbb{T}^n_\kappa$. If H was previously queried on some input, then $\mathcal{D}$ looks up its entry in L_{H} and returns its answer $\hat{c} \in \mathbb{T}^n_\kappa$. Otherwise, it returns the first unused $\hat{c}$ and updates the list.

Blind Signature Query. Upon receiving signature queries from the forger $\mathcal{A}$ as a user, $\mathcal{D}$ interacts as a signer with $\mathcal{A}$ according to the signing protocol. However, rather than computing $\hat{z}^*_{1,1}, \dots, \hat{z}^*_{\kappa,2}$ as described in Figure 3, $\mathcal{D}$ directly samples these elements from $D^n_{\mathbb{Z},s^*}$ and sends them back to the forger $\mathcal{A}$ with probability $\approx 1/M_{\mathcal{S}}$ (Lemma 2). The same applies for $\hat{z}_1, \hat{z}_2$ with probability $\approx 1/M_{\mathcal{U}}$. Hence, the signature is generated with probability $\approx 1/(M_{\mathcal{S}} \cdot M_{\mathcal{U}}) = 1/M$.

Output. After $k \leq q_{\mathsf{Sign}}$ successful executions of the signing protocol, $\mathcal{A}$ outputs $k+1$ distinct and valid pairs of messages and corresponding signatures $(\mu_1, \mathsf{sig}_1), \dots, (\mu_{k+1}, \mathsf{sig}_{k+1})$. Then, one of the following two cases applies:

Case 1. $\mathcal{D}$ finds two signatures of messages $\mu, \mu' \in \{\mu_1, \dots, \mu_{k+1}\}$ with the same $\hat{c}$. In this case the verification algorithm yields

$$\mathsf{H}(\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c} \pmod q), \tau', \tau) = \mathsf{H}(\hat{a}\hat{z}'_1 + \hat{z}'_2 - \hat{b}\hat{c} \pmod q), \nu', \nu) \ .$$

With overwhelming probability this implies that $\mu = \mu'$ and $\hat{a}\hat{z}_1 + \hat{z}_2 = \hat{a}\hat{z}'_1 + \hat{z}'_2 \mod q$ (otherwise, $\mathcal{A}$ would have found a second preimage of $\hat{c}$ or the binding property of Com does not hold). Since $\mu = \mu'$, this implies that $(\hat{z}_1, \hat{z}_2) \neq (\hat{z}'_1, \hat{z}'_2)$. This yields $\hat{a}(\hat{z}_1 - \hat{z}'_1) + (\hat{z}_2 - \hat{z}'_2) = 0 \pmod q$. Since $(\hat{z}_1, \hat{z}_2) \neq (\hat{z}'_1, \hat{z}'_2)$, it must be that $\hat{z}_1 \neq \hat{z}'_1$ or $\hat{z}_2 \neq \hat{z}'_2$. Therefore, w.l.o.g. it holds that $\hat{z}_1 \neq \hat{z}'_1$. Since the signatures are valid, we have $\|(\hat{z}_1, \hat{z}_2)\| \leq B$ and $\|(\hat{z}'_1, \hat{z}'_2)\| \leq B$. Hence, $\|(\hat{z}_1 - \hat{z}'_1, \hat{z}_2 - \hat{z}'_2)\| \leq 2B$.

Case 2. If all signatures output by $\mathcal{A}$ have distinct random oracle answers, then $\mathcal{D}$ guesses an index $i \in [k+1]$ such that $\hat{c}_i = \hat{c}_j$ for some $j \in [q_{\mathsf{H}}]$. Then, it records the pair $(\mu_i, (\tau', \mathbf{r}, \hat{z}_1, \hat{z}_2, \hat{c}_i))$ and invokes $\mathcal{A}$ again with the same random tape and random oracle queries $\{\hat{c}_1, \dots, \hat{c}_{j-1}, \hat{c}'_j, \dots, \hat{c}'_{q_{\mathsf{H}}}\}$, where $\{\hat{c}'_j, \dots, \hat{c}'_{q_{\mathsf{H}}}\}$ are fresh random elements. After the second invocation, the output of $\mathcal{A}$ (by assumption) includes a pair $(\mu'_i, (\tau'', \mathbf{r}'', \hat{z}'_1, \hat{z}'_2, \hat{c}'_i))$. By the General Forking Lemma [BN06] we have $\hat{c}_i \neq \hat{c}'_i$ with probability $\varepsilon_{\mathsf{fork}}$ (see below). Therefore, with overwhelming probability (binding and second preimage security) we have $\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c}_i = \hat{a}\hat{z}'_1 + \hat{z}'_2 - \hat{b}\hat{c}'_i \pmod q$. Thus, we obtain

$$\hat{a}(\hat{z}_1 - \hat{z}'_1) + (\hat{z}_2 - \hat{z}'_2) = \hat{b}(\hat{c}_i - \hat{c}'_i) \pmod q \ .$$

Since both signatures are valid, we have $\|(\hat{z}_1, \hat{z}_2)\| \leq B$ and $\|(\hat{z}'_1, \hat{z}'_2)\| \leq B$. This implies that $\|(\hat{z}_1 - \hat{z}'_1, \hat{z}_2 - \hat{z}'_2)\| \leq 2B$. Moreover we have $\|(\hat{c}_i - \hat{c}'_i)\|_\infty \leq 2$.

The reduction $\mathcal{D}$ retries at most q_{H}^{k+1} times with different random tape and random oracle queries.

Analysis. According to Lemma 2, simulating the computation of $\hat{z}^*_{1,1}, \dots, \hat{z}^*_{\kappa,2}$ by $\mathcal{D}$ (without having the secret key) is statistically indistinguishable from generating them as described in the protocol, and the simulation produces these elements with probability $\approx 1/M_{\mathcal{S}}$ as in a real execution.

Next, one of the $k+1$ pairs output by $\mathcal{A}$ is by assumption not generated during the execution of the signing protocol. The probability of correctly guessing the index i corresponding to this pair is $1/(k+1)$. The probability that $\hat{c}_i$ was a random oracle query made by $\mathcal{A}$ is $1 - 1/|\mathbb{T}^n_\kappa|$, where $|\mathbb{T}^n_\kappa| = 2^\kappa \binom{n}{\kappa}$.

Table 3. A prefix-free encoding due to [DLL+17] for the high-order bits of an integer $z \pmod q = z_1 \cdot 2^\tau + z_0$ distributed according to $D_{\mathbb{Z},\sigma}$, where $\sigma \approx 2^\tau$.

Integer	0	1	-1	$k \geq 2$	$-k \leq -2$
Representation	00	01	10	$110^{2k-4}1$	$110^{2k-3}1$
Bits	2	2	2	$2k-1$	$2k$

Thus, the probability that $\hat{c}_i = \hat{c}_j$ is $\varepsilon_\mathcal{A} - 1/|\mathbb{T}^n_\kappa|$. Furthermore, with probability $1/2$, one of the q_H^{k+1} runs of $\mathcal{A}$ yields the map $\{(i,j) : \; \hat{c}_i = \hat{c}_j\}$. According to the General Forking Lemma, the probability that $\hat{c}_i \neq \hat{c}'_i$ and $\hat{c}'_i$ is used by $\mathcal{A}$ in the forgery is at least $\varepsilon_\mathsf{fork} \geq \left(\varepsilon_\mathcal{A} - \frac{1}{|\mathbb{T}^n_\kappa|}\right) \cdot \left(\frac{\varepsilon_\mathcal{A} - 1/|\mathbb{T}^n_\kappa|}{q_\mathsf{Sign} + q_\mathsf{H}} - \frac{1}{|\mathbb{T}^n_\kappa|}\right)$. Therefore, the success probability of $\mathcal{D}$ is given by $\varepsilon_\mathcal{D} \geq \dfrac{\varepsilon_\mathsf{fork}}{2(k+1)}$, which is non-negligible if $\varepsilon_\mathcal{A}$ is non-negligible.

Finally, we analyze the case that users can generate a valid signature after an aborted interaction with the signer. Together with the elements $\hat{y}_1, \ldots, \hat{y}_\kappa$, $\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2}$, and $\hat{c}^*_1, \ldots, \hat{c}^*_\kappa$, the output $\mathsf{result} = (\tau, \rho, \mathbf{r}', \hat{p}_1, \ldots, \hat{p}_\kappa, \hat{e}_1, \hat{e}_2, \hat{c})$ of an aborted interaction satisfies the 3 checks carried out by $\mathcal{S}$ in the last step (see Figure 3). In the following we denote these checks by C1, C2, and C3. Now, assume that a user $\mathcal{U}$ obtains a valid signature $(\tau'', \mathbf{r}'', \hat{z}'_1, \hat{z}'_2, \hat{c}')$ from an aborted interaction. If $\hat{c}' = \hat{c}$, then by C2 we obtain $\hat{a}(\hat{z}_1 - \hat{z}'_1) + \hat{z}_2 - \hat{z}'_2 = 0 \pmod q$. The case $\hat{z}_1 = \hat{z}'_1$ contradicts C3, hence $\hat{z}_1 \neq \hat{z}'_1$. Note that $\|(\hat{z}_1, \hat{z}_2)\| \leq B + \eta s^* \sqrt{2\kappa n} = B + s/\alpha$, hence $\|(\hat{z}_1 - \hat{z}'_1, \hat{z}_2 - \hat{z}'_2)\| \leq 2B + s/\alpha$. If $\hat{c}' \neq \hat{c}$, then $\mathcal{U}$ may hide $\hat{c}'$ in $\hat{c}^*_1, \ldots, \hat{c}^*_\kappa$. In this case we have $\hat{c}^*_j = \hat{p}_j^{-1} \hat{c}_j = \hat{p}_j^{'-1} \hat{c}'_j$ by C1, where $\hat{p}'_j \neq \hat{p}_j$ for all $j \in [\kappa]$. Hence, $\hat{p}_j^{-1} = \hat{p}_j^{'-1} \hat{c}'_j \hat{c}_j^{-1}$. Therefore, $\mathcal{U}$ must be able to predict the output of H in order to compute $\hat{p}_j^{-1}$. The success probability by an aborted interaction is at least $\varepsilon_\mathsf{abort} \geq \varepsilon_\mathcal{A}(1 - 1/|\mathbb{T}^n_\kappa|)$, which is non-negligible if $\varepsilon_\mathcal{A}$ is non-negligible. Therefore, the overall success probability of $\mathcal{D}$ is $\varepsilon_\mathcal{D} \geq \min\{\frac{\varepsilon_\mathsf{fork}}{2(k+1)}, \varepsilon_\mathsf{abort}\}$. □

Remark 2. As mentioned in Section 1.2, strong one-more unforgeability already implies strong honest-user unforgeability [SU17, Lemma 10]. Furthermore, the above proof assumes that $\hat{a}$ is given, while it is actually generated from a seed in order to save bandwidth by only storing the seed instead of the whole polynomial. Security with this assumption can be proven by the following simple reduction: Assuming the existence of an adversary $\mathcal{A}$ against BLAZE, we construct an adversary $\mathcal{B}$ against a variant of BLAZE with public key $(\hat{a}, \hat{b})$. By modeling the function Expand as a programmable random oracle, $\mathcal{B}$ chooses a random seed', reprograms $\mathsf{Expand}(\mathsf{seed}') = \hat{a}$, and invokes $\mathcal{A}$ on input $(\mathsf{seed}', \hat{b})$. The output of $\mathcal{B}$ is then the same forgery generated by $\mathcal{A}$.

4 Implementation

In this section we give some important details about the implementation of BLAZE. There are several aspects subject to optimization. We follow the protocol and provide some insights into our optimizations. First, the choice of the ring $R_q = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ for n a power of 2 allows for efficient NTT-based polynomial multiplication. It further offers a suitable set of signed permutation polynomials $\hat{\mathbb{T}}$. Due to our choice of $q = 2^{31} - 2^{17} + 1$, modular reduction works highly efficient according to [Sei18], however without the need for Barret reductions. From seed we directly generate the NTT representation of $\hat{a}$ as it is always used just in the context of multiplications. By this we save one NTT transformation. We further improve the running time by omitting bit-reversals during NTT transformations in accordance to [Sei18]. We use the framework [MW17] in order to efficiently generate discrete Gaussians of arbitrary size that are centered around zero. Effectively, we apply the NTT twice, i.e., when multiplying with $\hat{a}$. In the other cases, we do not need any multiplications at all. For instance, multiplication with elements $\hat{p} \in \hat{\mathbb{T}}$ requires just to rotate the respective polynomial and change the signs, if necessary.

Table 4. A review of parameters and sizes of keys and signatures for BLAZE.

Parameter	Description	Bounds
λ	security parameter	
n	dimension	power of 2
q	modulus	prime, $q = 1 \pmod{2n}$
σ	standard deviation (secret key)	$\sigma > 0$
κ	Hamming weight of H's output	$2^\kappa \binom{n}{\kappa} \geq 2^\lambda$
s^*	standard deviation (signer)	$s^* = \alpha^* \sqrt{\kappa}\, \\|(\hat{s}_1, \hat{s}_2)\\|$, $\alpha^* > 0$
s	standard deviation (signatures)	$s = \eta\alpha\sqrt{2\kappa n} s^*$, $\alpha, \eta > 0$, $\eta^{2n} \exp(n(1-\eta^2)) \leq 2^{-\lambda}$
M	number of repetitions	$M = M_\mathcal{S} \cdot M_\mathcal{U}$, $M_\mathcal{S} = \exp(\frac{12}{\alpha^*} + \frac{1}{2\alpha^{*2}})$, $M_\mathcal{U} = \exp(\frac{12}{\alpha} + \frac{1}{2\alpha^2})$
secret key size (bit)		$2n\lceil\log(t\sigma + 1)\rceil$, $2e^{-t^2/2} \leq 2^{-\lambda}$
public key size (bit)		$n\lceil\log q\rceil + \lambda$
signature size without compression (bit)		$\kappa(1 + \lceil\log n\rceil) + 2n\lceil\log(ts + 1)\rceil + 2\lambda$

For the inversion of a monomial $\hat{p} \in \hat{\mathbb{T}}$, we apply Lemma 3. Since elements $\hat{c}_i^*$ are also elements of $\hat{\mathbb{T}}$, multiplication essentially corresponds to a rotation as described before. Our random oracle H outputs random elements from the set $\mathbb{T}_\kappa^n$. We apply the "inside-out" version of the Fisher-Yates shuffle, which is perfectly suitable for this kind of distributions. For generating uniform random bits, we expand a seed of large enough entropy to the desired output length using Shake. For instance, we generate the NTT transformation of the polynomial $\hat{a}$ in this way. We also use Shake in combination with the Fisher-Yates shuffle as a random oracle in order to hash inputs of H to an element in $\mathbb{T}_\kappa^n$. For the verification step we compare the squared lengths of the polynomials with the squared bound B^2 rather than using square roots. Finally, we describe the implementation of (De)Compress. Gaussian integers are optimally represented via Huffman encoding as carried out for instance in [DDLL13, DLL+17]. We consider the simplified approach proposed in [DLL+17, Section B.5]. Let z be an integer distributed according to $D_{\mathbb{Z},\sigma}$. Then, z can be written as $z \pmod q = z_1 \cdot 2^\tau + z_0$, where $\sigma \approx 2^\tau$. The value z_0 is almost uniform and hence is left uncompressed, while z_1 is encoded using the prefix-free encoding proposed in [DLL+17, Table 3], which we review in Table 3. On average, representing z requires in total $\approx \tau + 2.25$ bits.

5 Concrete Parameters and Comparison

In this section we propose concrete parameters for BLAZE and compare our results with the previous blind signature schemes [BGSS17, PSM17, Rüc10]. We review the parameter description of BLAZE in Table 4. The table also shows the theoretical sizes of keys and signatures, which we explain first. We then describe our parameter selection and the methodology to estimate the security. We note that parameters for the scheme [BGSS17] and [PSM17, Rüc10] were selected targeting 100 and 102 bits of security, respectively. Therefore, we select our parameters targeting approximately the same security level, namely 128 bits. We also propose further parameters for 192 bits of security (paranoid). Benchmarking our parameters were carried out on an Intel Core i7-6500U, operating at 2.3 GHz and 8GB of RAM.

Sizes. The secret key consists of 2 polynomials with entries from $D_{\mathbb{Z},\sigma}$. By Lemma 1 these entries are bounded by $t\sigma$ with probability $1 - 2\exp(-t^2/2)$, where t is chosen such that this probability is at least $1 - 2^{-\lambda}$. Therefore, these polynomials require $2n\lceil\log(t\sigma + 1)\rceil$ bits. The public key consists of a polynomial from R_q and a seed of λ bits. Hence, it occupies $n\lceil\log q\rceil + \lambda$ bits. Finally, the signature

Table 5. Parameters for BLAZE targeting 128 and 192 bits of security. Sizes are given in KB.

λ	n	q	σ	κ	α^*	α	s^*	s	$M_{\mathcal{S}}$	$M_{\mathcal{U}}$	M	sk size	pk size	signature size
128	1024	$\approx 2^{31}$	0.5	16	20	25	2172.2	11796306	1.8	1.6	2.9	0.8	3.9	6.6
192	2048	$\approx 2^{31}$	1	22	12	20	4322.7	31142799.7	2.7	1.8	4.9	2.5	7.8	14.1

consists of two polynomials with entries from $D_{\mathbb{Z},s}$ in addition to a polynomial from $\mathbb{T}_\kappa^n$ and two strings of λ bits. Thus, its size is bounded by $\kappa(1 + \lceil \log n \rceil) + 2n\lceil \log(ts + 1)\rceil + 2\lambda$ bits.

Parameters. Table 5 shows the parameters selected for BLAZE. We give some insights of how these parameters were selected. We set $n = 1024$, which is a typical choice for lattice-based schemes targeting medium or high security levels. The modulus q is chosen large enough such that the underlying RSIS instance provides the desired security level. At the same time, q is also small enough such that the RLWE instance underlying the public key and with the associated standard deviation σ is also hard enough. We set κ such that the cardinality of $\mathbb{T}_\kappa^n$ is large enough for security. The parameters $\alpha^*, \alpha, M_{\mathcal{S}}$, and $M_{\mathcal{U}}$ are selected as carried out in regular signature schemes such as [DDLL13, DLL+17].

Security. We describe the methodology used to estimate the security of the proposed parameters. We considered the asymptotically best algorithms known to solve the underlying lattice problems with no memory restrictions. More precisely, we used the well known and widely used LWE estimator [APS15] (with commit-id 62b5edc on 2019-09-11) to measure the hardness of recovering the secret key. Furthermore, we considered the lattice reduction algorithm BKZ [SE94, CN11] to estimate the hardness of forging signatures. BKZ uses a solver for the shortest vector problem (SVP) in lattices of dimension b, where b is called the block size. The best known SVP solver [BDGL16] runs in time $\approx 2^{0.292b}$. Running BKZ with block size b on an n-dimensional lattice $\mathcal{L}$ takes time $8n2^{0.292b+16.4}$ [BDGL16, Alb17]. After calling BKZ we obtain a vector of length $\delta^n \cdot \det(\mathcal{L})^{1/n}$, where $\delta = \left(b \cdot (\pi b)^{\frac{1}{b}}/(2\pi e)\right)^{\frac{1}{2(b-1)}}$ [Che13]. By Theorem 3, forging a signature implies finding $(\hat{v}_1, \hat{v}_2, \hat{v}_3) \neq 0$ such that $\hat{a}\hat{v}_1 + \hat{v}_2 = \hat{v}_3\hat{b}$, where $\|(\hat{v}_1, \hat{v}_2)\| \leq 2B + s/\alpha$ and $\|\hat{v}_3\|_\infty \leq 2$. This amounts to solving RSIS for the matrix $(\hat{a}, 1, \hat{b})$ with norm bound $\beta = \sqrt{(2B + s/\alpha)^2 + 4n}$. Given β we determined δ by setting $\beta = \delta^n \cdot \det(\mathcal{L})^{1/n}$. Then we used the formula of δ given above to deduce the minimum block size b required for BKZ to achieve δ. Then we computed the cost of BKZ.

Comparison. Table 1 shows that BLAZE significantly improves upon the schemes [BGSS17, PSM17, Rüc10] with respect to all relevant efficiency metrics and considerably large improvement factors. We note that we considered only the best parameter set proposed for RBS in [Rüc10, Table 3] for the target security level of 102 bits.

6 Conclusion

We highlight few notable conclusions from our results and possible future work. We presented BLAZE, a new practical lattice-based blind signature scheme providing statistical blindness under adversely-chosen keys [ANN06] and the strongest version of unforgeability [SU17] in the ROM. We have shown that BLAZE improves upon all previous works on blind signatures based on assumptions conjectured to withstand quantum computer attacks.

Similar to RBS, the unforgeability proof of BLAZE requires the signing queries q_{Sign} to be limited to $o(\lambda)$. As mentioned in [Rüc10] and originally by Pointcheval and Stern [PS00], this constraint is an artifact of the proof and is not unusual for efficient blind signatures. It was left open to achieve a polynomial-time reduction in both q_{Sign} and key size. We extend this research question to investigating the security of BLAZE in the quantum random oracle model (QROM). A possible direction towards this goal may involve the results of Kiltz et al. [KLS18] on the security of Fiat-Shamir signatures in QROM. For instance, the security in QROM may be obtained by considering a variant of BLAZE whose underlying identification scheme admits lossy public keys as defined in [AFLT12] (see [KLS18] for further details). Further improvements that can be made on BLAZE's design are the following:

- Adapt the compression technique of Bai and Galbraith [BG14] such that signatures consist of only one Gaussian polynomial $\hat{z}_1$ rather than a pair $(\hat{z}_1, \hat{z}_2)$. However, this approach requires further security analysis, since the property of *strong* one-more unforgeability is then not directly preserved. Consequently, the security of the resulting scheme under the new security model by Schröder and Unruh [SU17] cannot be established in a straightforward way.
- Reduce the complexity of BS.Sign by compressing all Gaussian elements exchanged using the algorithm Compress, i.e., by compressing $\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2}$ from the 3rd move and $\hat{e}_1, \hat{e}_2$ included in the proof of failure.
- Modify BLAZE so that its security is based on the module version of SIS and LWE [LS15]. This allows for more flexibility when selecting parameters.
- Finally, we note that by modifying BLAZE so that key recovery is based on RSIS rather than RLWE, it can directly be transformed into an identity-based blind signature scheme. Secret keys can then be extracted from the master secret key using any preimage sampleable trapdoor function, e.g., due to [MP12].

References

AFLT12. Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In *Advances in Cryptology–EUROCRYPT 2012*, pages 572–590. Springer, 2012. 16

AG18. Seres István András and János Gulácsy. A blind-signature-based e-voting platform on Ethereum. https://github.com/seresistvanandras/evoting, 2018. Accessed Sep 11, 2019. 1

Alb17. Martin R Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In *Advances in Cryptology–EUROCRYPT 2017*, pages 103–129. Springer, 2017. 15

ANN06. Michel Abdalla, Chanathip Namprempre, and Gregory Neven. On the (im)possibility of blind message authentication codes. In *Topics in Cryptology - CT-RSA 2006*, pages 262–279. Springer, 2006. 6, 11, 15

APS15. Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015. https://bitbucket.org/malb/lwe-estimator/src. 15

BDGL16. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, pages 10–24. SIAM, 2016. 15

BF11. Dan Boneh and David Mandell Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In *Public Key Cryptography - PKC 2011*, pages 1–16. Springer, 2011. 11

BG14. Shi Bai and Steven D Galbraith. An improved compression technique for signatures based on learning with errors. In *Cryptographers' Track at the RSA Conference*, pages 28–47. Springer, 2014. 16

BGSS17. Olivier Blazy, Philippe Gaborit, Julien Schrek, and Nicolas Sendrier. A code-based blind signature. In *IEEE International Symposium on Information Theory, ISIT 2017*, pages 2718–2722. IEEE, 2017. 2, 3, 4, 5, 14, 15

BL13. Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In *ACM Conference on Computer and Communications Security - CCS 13*, pages 1087–1098. ACM, 2013. 1

BN06. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *ACM conference on Computer and Communications Security*, pages 390–399. ACM, 2006. 12

CCT+11. Liang Chen, Yongquan Cui, Xueming Tang, Dongping Hu, and Xin Wan. Hierarchical id-based blind signature from lattices. In *International Conference on Computational Intelligence and Security, CIS 2011*, pages 803–807. IEEE Computer Society, 2011. 4, 18, 20

Cha82. David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology–CRYPTO 82*, pages 199–203, 1982. 1

Che13. Yuanmi Chen. *Réduction de réseau et sécurité concrete du chiffrement completement homomorphe*. PhD thesis, ENS-Lyon, France, 2013. 15

CN11. Yuanmi Chen and Phong Q Nguyen. BKZ 2.0: Better lattice security estimates. In *Advances in Cryptology–ASIACRYPT 2011*, pages 1–20. Springer, 2011. 15

CNS07. Jan Camenisch, Gregory Neven, and Abhi Shelat. Simulatable adaptive oblivious transfer. In *Advances in Cryptology–EUROCRYPT 2007*, pages 573–590. Springer, 2007. 11

Coo10. Howard A. Schmidt (National Cybersecurity Coordinator). National strategy for trusted identities in cyberspace. Cyberwar Resources Guide, Item #163, 2010. http://www.projectcyw-d.org/resources/items/show/163 (Accessed Sep. 11, 2019). 1

DDLL13. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians. In *Advances in Cryptology–CRYPTO 2013*, pages 40–56. Springer, 2013. 4, 14, 15

DKL+18. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *Transactions on Cryptographic Hardware and Embedded Systems - TCHES*, 2018(1):238–268, 2018. 3, 4

DLL+17. Leo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. CRYSTALS–Dilithium: Digital signatures from module lattices. Cryptology ePrint Archive, Report 2017/633, 2017. Version: 20170627:201152, http://eprint.iacr.org/2017/633. 13, 14, 15

FS09. Marc Fischlin and Dominique Schröder. Security of blind signatures under aborts. In *Public Key Cryptography - PKC*, pages 297–316. Springer, 2009. 11

Gem11. Gemalto. Integration of gemalto's smart card security with microsoft u-prove. https://www.securetechalliance.org/gemalto-integrates-smart-card-security-with-microsoft-u-prove, 2011. Accessed Sep. 11, 2019. 1

GHW+17. Wen Gao, Yupu Hu, Baocang Wang, Jia Xie, and Momeng Liu. Identity-based blind signature from lattices. *Wuhan University Journal of Natural Sciences*, 22(4):355–360, 2017. 4, 18, 20

GHWX16. Wen Gao, Yupu Hu, Baocang Wang, and Jia Xie. Identity-based blind signature from lattices in standard model. In *Information Security and Cryptology - Inscrypt 2016*, pages 205–218. Springer, 2016. 4, 18, 20

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Annual ACM symposium on Theory of Computing*, pages 197–206. ACM, 2008. 20

HBG16. Ethan Heilman, Foteini Baldimtsi, and Sharon Goldberg. Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions. In *Financial Cryptography and Data Security - FC 2016*, pages 43–60. Springer, 2016. 2

JLO97. Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures. In *Advances in Cryptology - CRYPTO 1997*, pages 150–164. Springer, 1997. 6

KKS17. Mahender Kumar, Chittaranjan Padmanabha Katti, and Prem Chandra Saxena. A secure anonymous e-voting system using identity-based blind signature scheme. In *International Conference on Information Systems Security, ICISS 2017*, pages 29–49. Springer, 2017. 1

KLS18. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. In *Advances in Cryptology–EUROCRYPT 2018*, pages 552–586. Springer, 2018. 16

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs Codes Cryptography*, 75(3):565–599, 2015. 16

Lyu09. Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology–ASIACRYPT 2009*, pages 598–616. Springer, 2009. 2

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In *Advances in Cryptology–EUROCRYPT 2012*, pages 738–755. Springer, 2012. 7

Mic07. Microsoft. Microsoft's open specification promise. https://docs.microsoft.com/en-us/openspecs/dev_center/ms-devcentlp/1c24c7c8-28b0-4ce1-a47d-95fe1ff504bc, 2007. Accessed Sep. 11, 2019. 1

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology - EUROCRYPT 2012*, pages 700–718. Springer, 2012. 16

MW17. Daniele Micciancio and Michael Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. In *Advances in Cryptology - CRYPTO 2017*, pages 455–485. Springer, 2017. 13

Paq13. Christian Paquin. U-Prove technology overview v1.1 (revision 2), 2013. https://www.microsoft.com/en-us/research/publication/u-prove-technology-overview-v1-1-revision-2/. 1, 2

PotEU01. European Parliament and Council of the European Union. Regulation (ec) no 45/2001. *Official Journal of the European Union*, 2001. 1

PotEU09. European Parliament and Council of the European Union. Directive 2009/136/ec. *Official Journal of the European Union*, 2009. 1

PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000. 3, 6, 15

PSM17. Albrecht Petzoldt, Alan Szepieniec, and Mohamed Saied Emam Mohamed. A practical multivariate blind signature scheme. In *Financial Cryptography and Data Security - FC 2017*, pages 437–454. Springer, 2017. 2, 3, 4, 5, 14, 15

Rüc10. Markus Rückert. Lattice-based blind signatures. In *Advances in Cryptology–ASIACRYPT 2010*, pages 413–430. Springer, 2010. 2, 3, 5, 14, 15, 18, 19, 20

SE94. Claus-Peter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66:181–199, 1994. 15

Sei18. Gregor Seiler. Faster AVX2 optimized NTT multiplication for Ring-LWE lattice cryptography. Cryptology ePrint Archive, Report 2018/039, 2018. http://eprint.iacr.org/2018/039. 13

SU17. Dominique Schröder and Dominique Unruh. Security of blind signatures revisited. *Journal of Cryptology*, 30(2):470–494, 2017. 3, 13, 15, 16

ZH16. Yanhua Zhang and Yupu Hu. Forward-secure identity-based shorter blind signature from lattices. *American Journal of Networks and Communications*, 5(2):17–26, 2016. 4, 18, 20

ZM14. Lili Zhang and Yanqin Ma. A lattice-based identity-based proxy blind signature scheme in the standard model. *Mathematical Problems in Engineering*, 2014, 2014. 4, 18, 20

ZTZ+17. Hongfei Zhu, Yu-an Tan, Xiaosong Zhang, Liehuang Zhu, Changyou Zhang, and Jun Zheng. A round-optimal lattice-based blind signature scheme for cloud services. *Future Generation Computer Systems*, 73:106–114, 2017. 4, 18

A The Blind Signature Scheme by Rückert

In this section we review the blind signature scheme proposed by Rückert [Rüc10]. For any positive integer x we write $R_{q,x}$ to denote the subset of the ring R_q consisting of all polynomials with coefficients in the set $\{-x, \ldots, 0, \ldots, x\}$. The scheme uses a random oracle $\mathsf{H} : \{0,1\}^* \to R_{q,1}$ and a commitment function $\mathsf{Com} : \{0,1\}^* \times \{0,1\}^n \to \{0,1\}^n$ that is statistically hiding and computationally binding. Key generation, signing, and verification are described in Figure 4.

B Cryptanalysis of other Lattice-Based Blind Signature Schemes

In this section we show how a user can simply compute the secret key of the lattice-based blind signature scheme given in [ZTZ+17]. Furthermore, we explain how the underlying SIS problem of all earlier identity-based (ID-based) blind signature proposals [CCT+11,ZM14,ZH16,GHWX16,GHW+17] can be solved by a user due to a design flaw.

B.1 Key Recovery of a Blind Signature Proposal

We describe a key recovery attack on a blind signature scheme proposed in [ZTZ+17]. We sketch its key generation and signing protocol and only explain the elements required for our analysis.

```
$\mathsf{BS.KGen}(1^\lambda)$
$\hat{a}_1, \ldots, \hat{a}_m \leftarrow_\$ R_q$
$\hat{s}_1, \ldots, \hat{s}_m \leftarrow_\$ R_{q,d}$
$\hat{b} \leftarrow \sum_1^m \hat{a}_i \hat{s}_i \pmod q$
$\mathsf{sk} := (\hat{s}_1, \ldots, \hat{s}_m)$
$\mathsf{pk} := (\hat{a}_1, \ldots, \hat{a}_m, \hat{b})$
```

```
$\mathsf{BS.Verify}(\hat{a}_1, .., \hat{a}_m, \hat{b}, \mu, (\mathbf{r}, \hat{z}_1, .., \hat{z}_m, \hat{c}))$
if $\hat{z}_i \in R_{q,z}$ for all $i \in [m]$ $\wedge$ $\hat{c} =$
    $\mathsf{H}\big(\sum_1^m \hat{a}_i \hat{z}_i - \hat{b}\hat{c} \pmod q, \mathsf{Com}(\mu; \mathbf{r})\big)$
    return 1
return 0
```

```
$\mathsf{BS.Sign}(\hat{s}_1, \ldots, \hat{s}_m, \hat{a}_1, \ldots, \hat{a}_m, \mu)$

Signer $\mathcal{S}(\hat{s}_1, \ldots, \hat{s}_m)$                User $\mathcal{U}(\hat{a}_1, \ldots, \hat{a}_m, \hat{b}, \mu)$
$\hat{y}_1^*, \ldots, \hat{y}_m^* \leftarrow_\$ R_{q,y}$
$\hat{y}^* \leftarrow \sum_1^m \hat{a}_i \hat{y}_i^* \pmod q$
        $\xrightarrow{\hat{y}^*}$
                                                    $\mathbf{r} \leftarrow_\$ \{0,1\}^n$, $C \leftarrow \mathsf{Com}(\mu; \mathbf{r})$
                                                    $\hat{u} \leftarrow_\$ R_{q,n}$, $\hat{v}_1, \ldots, \hat{v}_m \leftarrow_\$ R_{q,v}$
                                                    $\hat{c} \leftarrow \mathsf{H}(\hat{y}^* - \hat{b}\hat{u} - \sum_1^m \hat{a}_i \hat{v}_i, C)$
                                                    $\hat{c}^* \leftarrow \hat{c} - \hat{u}$
                                                    if $\hat{c}^* \notin R_{q,n-1}$ start over a new $\hat{u}$
                                                            $\xleftarrow{\hat{c}^*}$
for $i = 1, \ldots, m$ :
    $\hat{z}_i^* \leftarrow \hat{y}_i^* + \hat{s}_i \hat{c}^*$
    if $\hat{z}_i^* \notin R_{q,z^*}$ restart
        $\xrightarrow{\hat{z}_1^*, \ldots, \hat{z}_m^*}$
                                                    for $i = 1, \ldots, m$ :
                                                        $\hat{z}_i \leftarrow \hat{z}_i^* - \hat{v}_i$
                                                        if $\hat{z}_i \notin R_{q,z}$, result $\leftarrow (C, \hat{u}, \hat{v}_i, \hat{c})$
                                                        else result $\leftarrow$ ok
                                                            $\xleftarrow{\text{result}}$
if result $\neq$ ok
    if $\Big(\hat{c}^* + \hat{u} = \hat{c} = \mathsf{H}(\hat{y}^* - \hat{b}\hat{u} - \sum_1^m \hat{a}_i \hat{v}_i \pmod q, C) \wedge$
        $\mathsf{H}(\sum_1^m \hat{a}_i(\hat{z}_i^* - \hat{v}_i) - \hat{b}\hat{c} \pmod q, C) = \hat{c} \wedge$
        $\hat{z}_i^* - \hat{v}_i \notin R_{q,z}$ for some $i \in [m]\Big)$ restart
                                                    return $(\mu, (\mathbf{r}, \hat{z}_1, \ldots, \hat{z}_m, \hat{c}))$
```

Fig. 4. A formal description of the blind signature scheme by Rückert [Rüc10].

The secret key is an $(n \times n)$-matrix $\mathbf{S}$ with coefficients from $\{-1, 0, 1\}$ and the verification key is two $(n \times n)$-matrices $(\mathbf{P}, \mathbf{H})$, where $\mathbf{P} = \lfloor 2\rho(\mathbf{S}) + 1 \rfloor \cdot \mathbf{I}_n$, $\rho(\mathbf{S})$ is the spectral radius of $\mathbf{S}$, and $\mathbf{H}$ is the Hermite normal form of $\mathbf{P} - \mathbf{S}$. Signing is performed as follows:

1. The user sends an n-dimensional vector $\mathbf{u}$ to the signer, where $\mathbf{u}$ contains the message being signed and some random elements.
2. The signer sends $\mathbf{z}' = \mathbf{u} - \lfloor \mathbf{u}\mathbf{P}^{-1} \rceil (\mathbf{P} - \mathbf{S})$ back to the user.
3. The user outputs $\mathbf{z} = \mathbf{z}'\mathbf{T}^{-1} - \mathbf{e}$ as a part of the signature, where $\mathbf{T}, \mathbf{e}$ are included in $\mathbf{u}$.

The secret key $\mathbf{S}$ can be computed as follows. The user selects two random vectors $\mathbf{u}_1, \mathbf{u}_2$ such that $\mathbf{x} = \lfloor \mathbf{u}_1 \mathbf{P}^{-1} \rceil - \lfloor \mathbf{u}_2 \mathbf{P}^{-1} \rceil$ is invertible and initiates the signing protocol twice by sending $\mathbf{u}_1, \mathbf{u}_2$, respectively. After receiving $\mathbf{z}'_1, \mathbf{z}'_2$, the secret key is then given by $\mathbf{S} = (\mathbf{z}'_1 - \mathbf{z}'_2 - \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{x}\mathbf{P}) \cdot \mathbf{x}^{-1}$.

B.2 Forgeability of Earlier ID-Based Blind Signatures

We describe a design flaw in the previous identity-based blind signature schemes [CCT+11,ZM14,ZH16, GHWX16, GHW+17] which are based on lattices. They all follow the same framework to blindly sign messages. This framework employs the preimage sampleable trapdoor function[3] introduced in [GPV08]. It works as follows: Given a public random $(n \times m)$-matrix $\mathbf{A}$ with a short basis for the lattice $\Lambda_q^{\perp}(\mathbf{A}) = \left\{\mathbf{e} \in \mathbb{Z}^m \ : \ \mathbf{Ae} = \mathbf{0} \pmod q\right\}$ as a secret trapdoor. Signing is performed as follows.

1. The user sends an n-dimensional vector $\mathbf{y} = \mathbf{Ax} + \mathbf{c} \cdot t \pmod q$ to the signer, where $\mathbf{x}$ is an m-dimensional Gaussian vector, $\mathbf{c} \in \mathbb{Z}_q^n$ is the hash value of the message being signed, and t is a small integer.
2. The signer samples a preimage $\mathbf{e}$ such that $\mathbf{Ae} = \mathbf{y} \pmod q$ and sends it back to the user.
3. The user outputs the signature $\mathbf{z} = (\mathbf{e} - \mathbf{x}) \cdot t^{-1} \pmod q$. We note that two of the proposals we analyze here consider t as an invertible Gaussian $(n \times n)$-matrix, although the signature $\mathbf{z}$ cannot be obtained by multiplying an m-dimensional vector with an $(n \times n)$-matrix.

Verification is performed by checking that $\mathbf{Az} = \mathbf{c} \pmod q$. Apparently, it is assumed that the signing protocol is stateful in order to prevent re-querying attack. That is, the signer has a local storage for returning the same preimage of previous signing queries. Nevertheless, any user can simply send $\mathbf{y} = \mathbf{Ax} \pmod q$ and let the signer return a preimage $\mathbf{x}'$ of $\mathbf{y}$. Thus, we obtain $\mathbf{A}(\mathbf{x} - \mathbf{x}') = \mathbf{0} \pmod q$, where $\mathbf{x} - \mathbf{x}'$ is short and non-zero vector with high probability, since collisions always exist.

[3] We note that Rückert [Rüc10] already pointed out that blind signatures cannot be implemented using this trapdoor function.