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 current systems based on the hardness of classical security assumptions such as factoring and computing discrete logarithms. 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. Our software implementation 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 time of both key generation and verification is in the same order as state-of-the-art regular signature schemes, however several orders of magnitude 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 [10] and have become fundamental building blocks in privacy-oriented cryptography. One of the main applications of blind

signatures is anonymous credentials [5], 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 [31,32] and the National Strategy for Trusted Identities in Cyberspace [14]. An established real-life use case of blind signatures in anonymous credentials is the U-Prove technology [30] designed by Microsoft. U-Prove is one of the technologies, to which the Microsoft's Open Specification Promise [29] applies and is integrated for example by Gemalto - a leading digital security company - in its smart card technology in order to enhance privacy [20]. Another application of blind signatures is e-voting systems [24], where authorities blindly sign public keys used by voters to anonymously cast their votes. Further applications of blind signatures include e-cash systems utilizing the Bitcoin blockchain [21], where entities blindly sign digital coins withdrawn by users for selling and buying products and services over the Internet and open networks.

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 factoring large integers and computing 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 [30]. 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 [8, 33, 35], 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.

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 [26] 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 a software implementation of BLAZE attesting its practicality and propose parameters targeting 128 bits of security. Our implementation and parameters show that BLAZE is much more efficient than the previous blind signature schemes [8, 33, 35] 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. The parameters used in our implementation are in the order of current state-of-the-art ordinary signature

Table 1. Comparison of the existing blind signature schemes that are conjectured to be secure under quantum computer attacks. 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. Sizes are given in kilo bytes (KB), timings in milliseconds (ms) and cycles (in parentheses). Benchmarking our parameters were carried out on an Intel Core i7-6500U, operating at 2.3 GHz and 8GB of RAM.

Scheme	Bit security	Sizes			Performance		
		Secret key	Public key	Signature	Key generation	Signing	Verification
BLAZE (this work)	128	0.8	3.9	6.6	0.1 (204, 671)	17.8 (35, 547, 397)	0.1 (276, 210)
[35]	102	**23.6**	23.6	89.4	**52**	**283**	**57**
[33]	102	36.6	54.6	**17.6**	9392	3662	2656
[8]	100	-	**15**	200	-	-	-
Improvement factor		29.5	3.8	2.7	520	15.9	570

schemes such as the recent lattice-based NIST submission Dilithium [16]. 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. 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 [36, 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 [34].

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 [35] 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 the full version of this paper [3]. 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 applying rejection sampling on $\hat{c}^*$ to make sure that it masks $\hat{c}$. If this is not the case, $\mathcal{U}$ selects a new $\hat{u}$ and repeats until success and then proceeds by sending $\hat{c}^*$ to $\mathcal{S}$. Subsequently, $\mathcal{S}$ responds with elements $\hat{z}_1^*, \ldots, \hat{z}_m^*$ only after carrying out rejection sampling on this response and making sure that it does not leak information about the secret key, otherwise $\mathcal{S}$ restarts the protocol. Then, $\mathcal{U}$ transforms this response into the signature part $(\hat{z}_1, \ldots, \hat{z}_m)$. Here, $\mathcal{U}$ applies rejection sampling in order to further maintain blindness. More precisely, the polynomials $\hat{z}_i^*$ must be 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 en-

sure 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 schemes) in addition to removing the first rejection sampling procedure 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 rotation. 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 step, it also ensures shorter signatures and speeds up the rejection sampling process 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., [15,16]. 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 procedure 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.

Related work. In addition to RBS, there are other lattice-based constructions of blind signatures found in literature. However, we show in the full version of this paper [3] that they are unfortunately insecure. More precisely, we show for the proposal in [39] how the secret key can simply be recovered already after two executions of its signing protocol. For the remaining schemes [11,18,19,37,38]

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 [33] and the code-based one proposed in [8]. Table 1 shows that BLAZE is much more efficient than those schemes in terms of all efficiency metrics.

Outline. In Section 2 we give the background required throughout this work. In Section 3 we present our new blind signature scheme BLAZE. In Section 4 we propose concrete parameters and compare BLAZE with the schemes [8, 33, 35]. We conclude our results and discuss possible future directions in Section 5.

2 Preliminaries

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. The norms of $\hat{\mathbf{a}}$ are 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 $\mathrm{negl}(\lambda)$ we denote a negligible function in λ. A probability is called overwhelming if it is at least $1 - \mathrm{negl}(\lambda)$. The *statistical distance* between two distributions X, Y over a countable domain D is defined by $\frac{1}{2}\sum_n |X(n) - Y(n)|$. 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.*

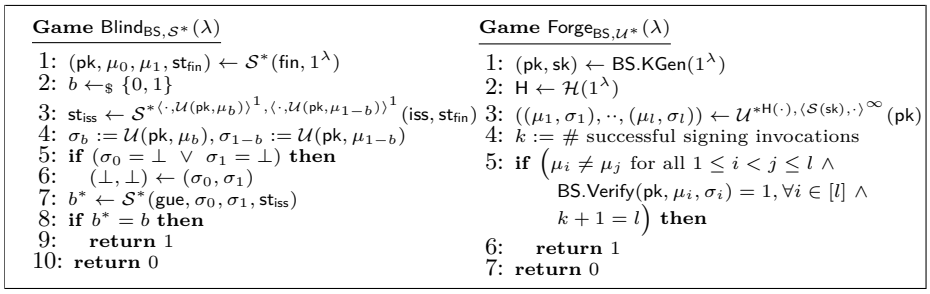

Fig. 1. Security games of blindness and one-more unforgeability. In the blindness game the modes find, issue, guess are shortened to fin, iss, gue, respectively.

- *BS.Sign*$(\mathsf{sk}, \mathsf{pk}, \mu)$ *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* $\sigma = \bot$ *to denote failure.*
- *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., 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 [22, 34].

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 1 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* $\varepsilon = \mathrm{Adv}_{\mathsf{BS},\mathcal{S}^*}(\lambda) = \big|\Pr[b^* = b] - \frac{1}{2}\big|$*. The scheme is statistically blind if it is* $(t = \infty, \varepsilon = \mathrm{negl}(\lambda))$*-blind.*

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 1 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* $1 \leq i < j \leq l$*.*

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

$\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^k\} \subset \mathbb{R}^m$. 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. Sampling from $D_{\mathcal{L},\sigma}$ using a specified randomness ρ is denoted by $D_{\mathcal{L},\sigma}(\rho)$. 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 ([27, 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 ([27, 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. In the following 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 asks to find a non-zero vector $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_{k+1}) \in R^{k+1}$ such that $[\hat{\mathbf{a}}\ \mathbf{1}] \cdot \hat{\mathbf{x}} = 0 \pmod q$ and $\|\hat{\mathbf{x}}\| \leq \beta$. The inhomogeneous RSIS problem 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)). *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 $\mathcal{S}$ to conceal the secret key and one by the user $\mathcal{U}$ 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. The proofs are provided in the full version of this paper [3].

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 in* R *and the inverse of* $\hat{p}$ *is given by* $\hat{p}^{-1} = (-1)^{1-s} \cdot x^{n-i} \in \hat{\mathbb{T}}$.

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

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

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

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 generator). It takes a random seed as input and expands it to any desired length. This function is solely used for saving bandwidth as it is deterministic, i.e., given some input it always produces the same output. We let H be a public hash function modeled as a random oracle and randomly chosen from the family $\{\mathsf{H} : \{0,1\}^* \to \mathbb{T}_\kappa^n\}$. We further let $\mathsf{Com} : \{0,1\}^* \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a statistically hiding and computationally binding commitment function. Finally, we let Compress and Decompress be functions for (de)compressing Gaussian elements (see the full version [3] for description). The respective algorithms of BLAZE are formally described in Figure 2.

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

Signing. Given sk, seed, and a message μ, $\mathcal{S}$ samples 2κ masking terms $\hat{y}_{j,1}^*, \hat{y}_{j,2}^*$ from $D_{\mathbb{Z},s^*}^n$ for $j \in [\kappa]$ and sends $\hat{y}_j = \hat{a}\hat{y}_{j,1}^* + \hat{y}_{j,2}^* \pmod q$ to $\mathcal{U}$.

Upon receiving $\hat{y}_1, \ldots, \hat{y}_\kappa$, $\mathcal{U}$ computes the commitments $\tau = \mathsf{Com}(\mu; \mathbf{r})$, $\tau' = \mathsf{Com}(\rho'; \mathbf{r}')$ for uniformly random selected $\mathbf{r}, \mathbf{r}', \rho'$ from $\{0,1\}^\lambda$, expands seed to the polynomial $\hat{a}$ using the function Expand, and selects uniformly random elements $\hat{p}_1, \ldots, \hat{p}_\kappa \in \hat{\mathbb{T}}$. Furthermore, $\mathcal{U}$ samples two polynomials $\hat{e}_1, \hat{e}_2$ from $D^n_{\mathbb{Z},s}$ using a randomness $\rho \in \{0,1\}^\lambda$, which is used to reduce the communication complexity, i.e., a proof of failure sent by $\mathcal{U}$ (see below) includes only ρ rather than $\hat{e}_1, \hat{e}_2$. 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. Afterwards, $\mathcal{U}$ masks each partition $\hat{c}_j$ by computing $\hat{c}^*_j = \hat{p}_j^{-1} \cdot \hat{c}_j$ for all $j \in [\kappa]$. Then, $\mathcal{U}$ sends $\hat{c}^*_1, \ldots, \hat{c}^*_\kappa$ to $\mathcal{S}$.

Using the partitions $\hat{c}^*_j$, $\mathcal{S}$ 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, $\mathcal{S}$ applies rejection sampling on $\hat{z}^*_{j,1}, \hat{z}^*_{j,2}$ to make sure that they do not leak information about sk. If $\mathsf{RejSamp}$ outputs 1, $\mathcal{S}$ sends $\hat{z}^*_{j,1}, \hat{z}^*_{j,2}$ to $\mathcal{U}$, otherwise $\mathcal{S}$ restarts the protocol.

Upon receiving $\hat{z}^*_{j,1}, \hat{z}^*_{j,2}$ $(j \in [\kappa])$, $\mathcal{U}$ computes $\hat{v}_1 = \sum_1^\kappa \hat{p}_j \hat{z}^*_{j,1}$, $\hat{v}_2 = \sum_1^\kappa \hat{p}_j \hat{z}^*_{j,2}$ and checks that $\|(\hat{v}_1, \hat{v}_2)\|$ is bounded by $\eta s^* \sqrt{2\kappa n}$. This check rules out malicious signers and ensures that the generated signatures are valid and blind. This check can be skipped in applications with trustworthy signers. In order for the verification to succeed, the pair $(\hat{z}_1, \hat{z}_2)$ that will be 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 applying $\mathsf{RejSamp}(\hat{z}_1, \hat{z}_2; \rho')$ 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 input to H. 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 the function $\mathsf{Compress}$ and sends $\mathsf{result} = \mathsf{ok}$ to $\mathcal{S}$. The signature is given by the tuple $(\tau', \mathbf{r}, \hat{z}_1, \hat{z}_2, \hat{c})$. If $\mathsf{RejSamp}$ outputs 0, $\mathcal{U}$ sends $\mathcal{S}$ a proof of failure by setting $\mathsf{result} = (\tau, \rho, \rho', \mathbf{r}', \hat{p}_1, \ldots, \hat{p}_\kappa, \hat{c})$. This allows $\mathcal{S}$ to perform 3 checks (see Figure 2) in order to verify that $\mathcal{U}$ indeed has not obtained a valid signature, and hence restarts the protocol.

Note that in order to verify that the rejection sampling process applied on $\hat{z}_1, \hat{z}_2$ does not accept using some randomness, $\mathcal{S}$ requires the randomness ρ' used by $\mathcal{U}$ for which $\mathsf{RejSamp}(\hat{z}_1, \hat{z}_2; \rho') = 0$. Therefore, ρ' 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 $\mathcal{U}$ computes a commitment τ' to ρ' and includes τ' in the signature in addition to involving τ' in the computation of $\hat{c}$ 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 $\big(\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c} \pmod q, \tau', \mathsf{Com}(\mu; \mathbf{r})\big)$ is equal to $\hat{c}$.

The following states the completeness, blindness, and strong one-more unforgeability of BLAZE. The completeness proof is provided in the full version [3].

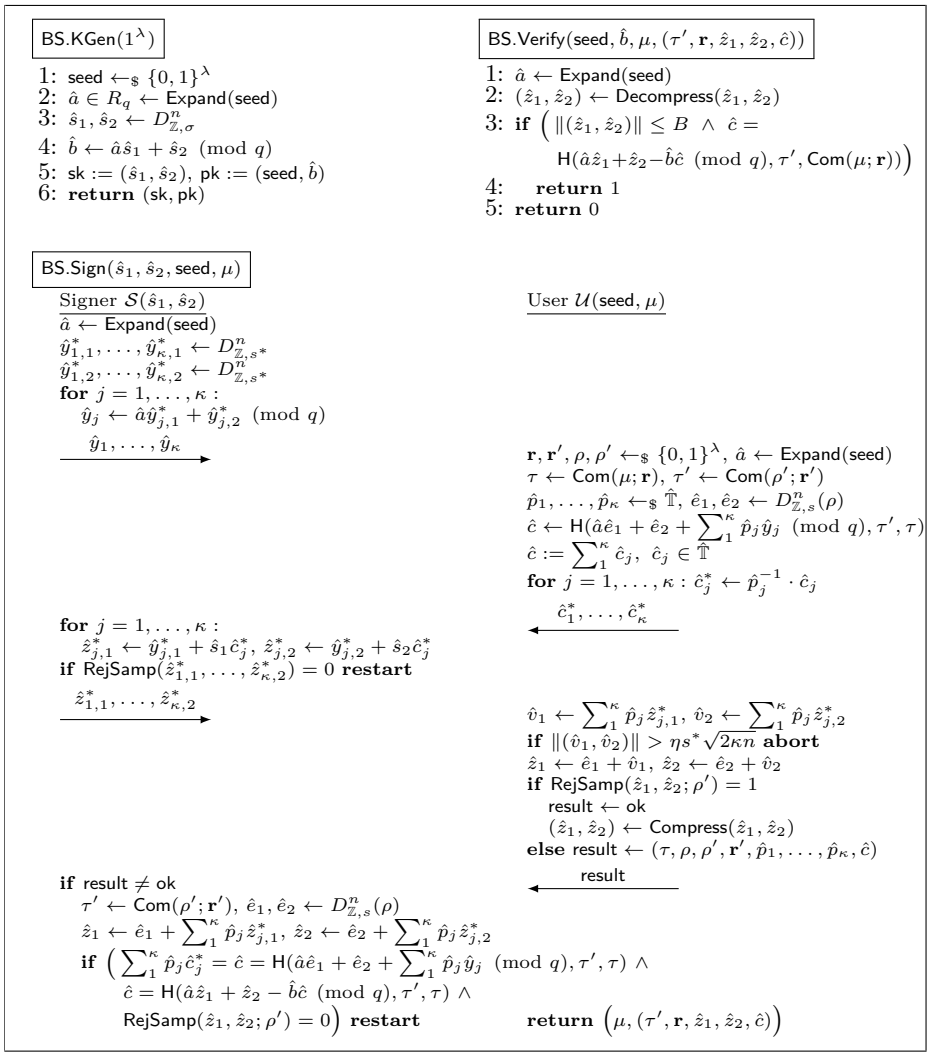

Fig. 2. A description of the new blind signature scheme BLAZE.

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.*

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 $\mathcal{U}$ 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 $\mathcal{U}$'s output are independently distributed, especially also from the message being signed. This requires analyzing only the pair $(\hat{z}_1, \hat{z}_2)$, since τ' is statistically hiding, $\mathbf{r}$ is uniformly random, $\hat{c} \in \mathbb{T}_\kappa^n$, and $\hat{c}_1^*, \ldots, \hat{c}_\kappa^* \in \hat{\mathbb{T}}$ are uniformly random and independently distributed by Lemma 4.

Let $(\hat{z}_1, \hat{z}_2)_b$, $(\hat{z}_1, \hat{z}_2)_{1-b}$ be the pairs output by $\mathcal{U}(\mathsf{seed}, \mu_b), \mathcal{U}(\mathsf{seed}, \mu_{1-b})$, 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 $D_{\mathbb{Z},s^*}^n$, 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 $D_{\mathbb{Z},s}^{2n}$.

Furthermore, if the protocol needs to be restarted, then the user selects fresh $\mathbf{r}, \mathbf{r}', \rho, \rho', \hat{p}_1, \ldots, \hat{p}_\kappa$. 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.

Finally, we note that checking the length of $(\hat{v}_1, \hat{v}_2)$ made by the user (see Figure 2) maintains blindness by preventing a malicious signer from choosing $(\hat{z}_{1,1}^*, \ldots, \hat{z}_{\kappa,2}^*)$ according to some distribution that makes the protocol fail. □

Remark 1. Similar to RBS, we note that BLAZE remains blind under the stronger blindness definition given in [1], 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 [9] is already achieved since a commitment to the message is being signed using a statistically hiding commitment scheme [17].

Recovering the secret key of BLAZE is as hard as RLWE. Thus, we prove its strong one-more unforgeability assuming the hardness of RLWE.

Theorem 3. *Let Com be a statistically hiding and computationally binding commitment function. 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 $\approx 1/M$, 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 uniform random $\hat{a} \in R_q$. The reduction $\mathcal{D}$ selects $\hat{s}_1, \hat{s}_2 \leftarrow D^n_{\mathbb{Z},\sigma}$ and computes $\hat{b} = \hat{a}\hat{s}_1 + \hat{s}_2 \pmod q$. Then, $\mathcal{D}$ randomly selects answers for random oracle queries $\{\hat{c}_1, \ldots, \hat{c}_{q_{\mathsf{H}}}\}$. 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 $(\hat{t}, \tau', \tau)$, 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 (see Figure 2). The elements $\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2}$ are output 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 $(\mu_1, \mathsf{sig}_1), \ldots, (\mu_{k+1}, \mathsf{sig}_{k+1})$. Then, one of the following two cases applies:

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

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

This implies that $\mu = \mu'$ and $\hat{a}\hat{z}_1 + \hat{z}_2 = \hat{a}\hat{z}'_1 + \hat{z}'_2 \pmod q$ with overwhelming probability (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 $\{\hat{c}_1, \ldots, \hat{c}_{j-1}, \hat{c}'_j, \ldots, \hat{c}'_{q_{\mathsf{H}}}\}$, where $\{\hat{c}'_j, \ldots, \hat{c}'_{q_{\mathsf{H}}}\}$ are fresh random elements. After the second invocation, the output of $\mathcal{A}$ includes a pair $(\mu'_i, (\tau'', \mathbf{r}'', \hat{z}'_1, \hat{z}'_2, \hat{c}'_i))$. Hence, $\mathcal{D}$ returns $(\hat{z}_1 - \hat{z}'_1, \hat{z}_2 - \hat{z}'_2, \hat{c}_i - \hat{c}'_i)$. The reduction $\mathcal{D}$ retries at most q_{H}^{k+1} times with different random tape and random oracle queries.

Analysis. First, we note that the environment of $\mathcal{A}$ is perfectly simulated by $\mathcal{D}$ and signatures are generated with the same probability as in the original execution of the protocol. If the first case (**Case 1.**) occurs, $\mathcal{D}$ solves RSIS for the matrix $(\hat{a}, 1)$, i.e., $\mathcal{D}$ finds $(\hat{v}_1, \hat{v}_2) \neq 0$ satisfying $\hat{a}\hat{v}_1 + \hat{v}_2 = 0 \pmod q$ and $\|(\hat{v}_1, \hat{v}_2)\| \leq 2B$. Next, we analyze the second case (**Case 2.**). In this case one of the $k+1$ pairs output by $\mathcal{A}$ is by assumption not generated during the execution of the 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 $1-1/|\mathbb{T}_\kappa^n|$. Thus, the probability that $\hat{c}_i = \hat{c}_j$ is $\varepsilon_\mathcal{A} - 1/|\mathbb{T}_\kappa^n|$. Furthermore, there are q_H^{k+1} index maps $\{(i,j) : \hat{c}_i = \hat{c}_j\}$. By the General Forking Lemma [7], we have $\hat{c}_i \neq \hat{c}'_i$, $\hat{a}\hat{z}_1 + \hat{z}_2 - \hat{b}\hat{c}_i = \hat{a}\hat{z}'_1 + \hat{z}'_2 - \hat{b}\hat{c}'_i \pmod q$, and $\hat{c}'_i$ is used by $\mathcal{A}$ in the forgery is at least $\varepsilon_\mathsf{fork} \geq \left(\varepsilon_\mathcal{A} - \frac{1}{|\mathbb{T}_\kappa^n|}\right) \cdot \left(\frac{\varepsilon_\mathcal{A} - 1/|\mathbb{T}_\kappa^n|}{q_\mathsf{Sign}+q_\mathsf{H}} - \frac{1}{|\mathbb{T}_\kappa^n|}\right)$. Thus, by setting $\hat{b} = \hat{a}\hat{s}_1 + \hat{s}_2 \pmod q$ we obtain

$$\hat{a}\hat{v}_1 + \hat{v}_2 = 0 \pmod q, \tag{2}$$

where $\hat{v}_1 = \hat{z}_1 - \hat{z}'_1 - \hat{s}_1(\hat{c} - \hat{c}')$ and $\hat{v}_2 = \hat{z}_2 - \hat{z}'_2 - \hat{s}_2(\hat{c} - \hat{c}')$. Since $\hat{s}_1, \hat{s}_2$ are not uniquely defined, $\mathcal{A}$ does not know which $\hat{s}_1, \hat{s}_2$ is being used to construct $\hat{v}_1, \hat{v}_2$, hence with probability at least $1/2$ we have $(\hat{v}_1, \hat{v}_2) \neq (0,0)$. By rearranging the terms in (2) 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 $\|(\hat{c}_i - \hat{c}'_i)\|_\infty \leq 2$. This constitutes a solution to inhomogeneous RSIS. 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. The proof of failure result satisfies the 3 checks carried out by $\mathcal{S}$ in the last step (see Figure 2). In the following we denote these checks by C1, C2, and C3. Now, assume that a user $\mathcal{U}$ obtains a valid signature $(\tau'', \mathbf{r}'', \hat{z}'_1, \hat{z}'_2, \hat{c}')$ after 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 $\varepsilon_\mathsf{abort} \geq \varepsilon_\mathcal{A}(1 - 1/|\mathbb{T}_\kappa^n|)$, 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, strong one-more unforgeability already implies strong honest-user unforgeability [36, Lemma 10].

4 Concrete Parameters and Comparison

In this section we propose parameters for BLAZE and compare our results with the previous blind signature schemes [8, 33, 35]. We review the parameter description of BLAZE and the sizes of keys and signatures in Table 2. We then describe our parameter selection and the methodology to estimate the security. We note that parameters for the scheme [8] and [33, 35] were selected targeting 100 and 102 bits of security, respectively. Therefore, we select our parameters

Table 2. A review of parameters and sizes of keys and signatures of 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)		$\kappa(1 + \lceil\log n\rceil) + 2n\lceil\log(ts+1)\rceil + 2\lambda$

targeting approximately the same security level, namely 128 bits. A description of our software implementation can be found in the full version of this paper [3].

Parameters. Table 3 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 [15].

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 [2] (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 [13] 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 [6] runs in time $\approx 2^{0.292b}$. Running BKZ with block size b on an n-dimensional lattice $\mathcal{L}$ takes time $8n2^{0.292b+16.4}$ [6]. After calling BKZ we obtain a vector of length $\delta^n \cdot \det(\mathcal{L})^{1/n}$, where $\delta = \left(b \cdot (\pi b)^{\frac{1}{b}}/(2\pi e)\right)^{\frac{1}{2(b-1)}}$ [12]. 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 $\beta = \sqrt{(2B + s/\alpha)^2 + 4\kappa}$. 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 previous schemes [8,33,35] with respect to all efficiency metrics and with considerably

Table 3. Parameters for BLAZE targeting 128 bits of security. Sizes are given in KB.

λ	n	q	σ	κ	α^*	α	s^*	s	M_S	M_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

large improvement factors. We note that we considered only the best parameter set proposed for RBS in [35, Table 3] for the target security level of 102 bits.

5 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 [1] and the strongest version of unforgeability [36] 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 [35] and originally by Pointcheval and Stern [34], 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. [23] 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 (see [23] for further details). Further improvements that can be made on BLAZE's design are the following:

- Adapt the compression technique of [4] such that signatures consist of only one Gaussian polynomial $\hat{z}_1$ rather than a pair $(\hat{z}_1, \hat{z}_2)$. This approach requires further analysis regarding correctness and security. Moreover, the *strong* one-more unforgeability is then not directly preserved. Consequently, the security of the resulting scheme under the new security model due to [36] cannot be established in a straightforward way.
- Reduce the communication complexity of BS.Sign by compressing the Gaussian elements $\hat{z}^*_{1,1}, \ldots, \hat{z}^*_{\kappa,2}$ using the algorithm Compress before sending.
- Modify BLAZE so that its security is based on the module version of SIS and LWE [25]. 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 [28].

Acknowledgements. The authors are grateful to the anonymous reviewers of FC20 for their comments and suggestions. This work has been partially supported by the German Research Foundation (DFG) as part of project P1 within the CRC 1119 CROSSING.

References

1. Abdalla, M., Namprempre, C., Neven, G.: On the (im)possibility of blind message authentication codes. In: Topics in Cryptology - CT-RSA 2006. pp. 262–279. Springer (2006)
2. Albrecht, M., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015), https://bitbucket.org/malb/lwe-estimator/src
3. Alkeilani Alkadri, N., El Bansarkhani, R., Buchmann, J.: BLAZE: Practical lattice-based blind signatures for privacy-preserving applications. Cryptology ePrint Archive, Report 2019/1167 (2019), http://eprint.iacr.org/2019/1167, Full version of this paper
4. Bai, S., Galbraith, S.: An improved compression technique for signatures based on learning with errors. In: CT-RSA 2014. pp. 28–47. Springer (2014)
5. Baldimtsi, F., Lysyanskaya, A.: Anonymous credentials light. In: ACM Conference on Computer and Communications Security - CCS 13. pp. 1087–1098. ACM (2013)
6. Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: ACM-SIAM Symposium on Discrete Algorithms, SODA 2016. pp. 10–24. SIAM (2016)
7. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: ACM conference on Computer and Communications Security. pp. 390–399. ACM (2006)
8. Blazy, O., Gaborit, P., Schrek, J., Sendrier, N.: A code-based blind signature. In: IEEE International Symposium on Information Theory, ISIT 2017. pp. 2718–2722. IEEE (2017)
9. Camenisch, J., Neven, G., Shelat., A.: Simulatable adaptive oblivious transfer. In: Advances in Cryptology–EUROCRYPT 2007, pp. 573–590. Springer (2007)
10. Chaum, D.: Blind signatures for untraceable payments. In: Advances in Cryptology–CRYPTO 82. pp. 199–203 (1982)
11. Chen, L., Cui, Y., Tang, X., Hu, D., Wan, X.: Hierarchical id-based blind signature from lattices. In: International Conference on Computational Intelligence and Security, CIS 2011. pp. 803–807. IEEE Computer Society (2011)
12. Chen, Y.: Réduction de réseau et sécurité concrete du chiffrement completement homomorphe. Ph.D. thesis, ENS-Lyon, France (2013)
13. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Advances in Cryptology–ASIACRYPT 2011, pp. 1–20. Springer (2011)
14. Coordinator), H.A.S.N.C.: National strategy for trusted identities in cyberspace. Cyberwar Resources Guide, Item #163 (2010), http://www.projectcywd.org/resources/items/show/163
15. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal Gaussians. In: Advances in Cryptology–CRYPTO 2013, pp. 40–56. Springer (2013)
16. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium: A lattice-based digital signature scheme. Transactions on Cryptographic Hardware and Embedded Systems - TCHES (1), 238–268 (2018)
17. Fischlin, M., Schröder, D.: Security of blind signatures under aborts. In: Public Key Cryptography - PKC. pp. 297–316. Springer (2009)
18. Gao, W., Hu, Y., Wang, B., Xie, J.: Identity-based blind signature from lattices in standard model. In: Information Security and Cryptology - Inscrypt 2016. pp. 205–218. Springer (2016)

19. Gao, W., Hu, Y., Wang, B., Xie, J., Liu, M.: Identity-based blind signature from lattices. Wuhan University Journal of Natural Sciences **22**(4), 355–360 (2017)
20. 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)
21. Heilman, E., Baldimtsi, F., Goldberg, S.: Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions. In: Financial Cryptography and Data Security - FC 2016. pp. 43–60. Springer (2016)
22. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures. In: Advances in Cryptology - CRYPTO 1997. pp. 150–164. Springer (1997)
23. Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. In: Advances in Cryptology–EUROCRYPT 2018. pp. 552–586. Springer (2018)
24. Kumar, M., Katti, C.P., Saxena, P.C.: A secure anonymous e-voting system using identity-based blind signature scheme. In: International Conference on Information Systems Security, ICISS 2017. pp. 29–49. Springer (2017)
25. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Designs Codes Cryptography **75**(3), 565–599 (2015)
26. Lyubashevsky, V.: Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: Advances in Cryptology–ASIACRYPT 2009, pp. 598–616. Springer (2009)
27. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Advances in Cryptology–EUROCRYPT 2012, pp. 738–755. Springer (2012)
28. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Advances in Cryptology - EUROCRYPT 2012, pp. 700–718. Springer (2012)
29. Microsoft: Microsoft's open specification promise. https://docs.microsoft.com/en-us/openspecs/dev_center/ms-devcentlp/1c24c7c8-28b0-4ce1-a47d-95fe1ff504bc (2007)
30. Paquin, C.: 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/
31. Parliament, E., of the European Union, C.: Regulation (ec) no 45/2001. Official Journal of the European Union (2001)
32. Parliament, E., of the European Union, C.: Directive 2009/136/ec. Official Journal of the European Union (2009)
33. Petzoldt, A., Szepieniec, A., Mohamed, M.S.E.: A practical multivariate blind signature scheme. In: Financial Cryptography and Data Security - FC 2017. pp. 437–454. Springer (2017)
34. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology **13**(3), 361–396 (2000)
35. Rückert, M.: Lattice-based blind signatures. In: Advances in Cryptology–ASIACRYPT 2010, pp. 413–430. Springer (2010)
36. Schröder, D., Unruh, D.: Security of blind signatures revisited. Journal of Cryptology **30**(2), 470–494 (2017)
37. Zhang, L., Ma, Y.: A lattice-based identity-based proxy blind signature scheme in the standard model. Mathematical Problems in Engineering **2014** (2014)
38. Zhang, Y., Hu, Y.: Forward-secure identity-based shorter blind signature from lattices. American Journal of Networks and Communications **5**(2), 17–26 (2016)
39. Zhu, H., Tan, Y., Zhang, X., Zhu, L., Zhang, C., Zheng, J.: A round-optimal lattice-based blind signature scheme for cloud services. Future Generation Computer Systems **73**, 106–114 (2017)