Lattice-based Blind Signatures

Markus Rückert*
markus.rueckert@cased.de

Technische Universität Darmstadt
Department of Computer Science
Cryptography and Computeralgebra
Germany

Abstract. Blind signatures (BS), introduced by Chaum, have become a cornerstone in privacy-oriented cryptography. Using hard lattice problems, such as the shortest vector problem, as the basis of security has advantages over using the factoring or discrete logarithm problems. For instance, lattice operations are more efficient than modular exponentiation and lattice problems remain hard for quantum and subexponential-time adversaries. Generally speaking, BS allow a signer to sign a message without seeing it, while retaining a certain amount of control over the process. In particular, the signer can control the number of issued signatures. For the receiver of the signature, this process provides perfect anonymity, e.g., his spendings remain anonymous when using BS for electronic money.
We provide a positive answer to the question of whether it is possible to implement BS based on lattice problems. More precisely, we show how to turn Lyubashevsky's identification scheme into a BS scheme, which has almost the same efficiency and security in the random oracle model. In particular, it offers quasi-linear complexity, statistical blindness, and its unforgeability is based on the hardness of worst-case lattice problems with an approximation factor of $\widetilde{\mathcal{O}}(n^5)$ in dimension n. Moreover, it is the first blind signature scheme that supports leakage-resilience, tolerating leakage of a $(1 - o(1))$ fraction of the secret key in a model that is inspired by Katz and Vaikuntanathan.

Keywords. Blind signatures, post-quantum, lattices, provable security, leakage resilience

*This work was supported by CASED (www.cased.de).

1. Introduction

Since Chaum proposed his idea of blind signatures [Cha82], it has become an important primitive for anonymous Internet banking, e-voting (e.g., [RHOAGZ07]), as well as for oblivious transfer [CNS07]. These applications will retain their importance in both, near and far future. As for the near future, we are convinced that current factoring and discrete logarithm based instantiations are efficient and secure. *But for how long?*

Today, when building provably secure cryptographic schemes, one also has to anticipate emerging technologies that may lead to new attacks. This is why we typically try to use the mildest possible assumptions. Let us consider the example of quantum computers as a metaphor for these future developments. In the quantum-age, the cryptographic assumptions change with the leap in computing power that quantum computers will provide. There are only a few cryptographic assumptions that are conjectured to be *post-quantum*, i.e., they are considered to withstand quantum computer attacks. One of those assumptions is the hardness of finding short vectors in a lattice. Even for today, there are benefits when building cryptography upon hard lattice problems because, unlike factoring, they withstand subexponential attacks and the best known algorithms, e.g., [AKS01], have an exponential complexity in the lattice dimension. Furthermore, lattice problems typically allow a worst-case to average-case reduction that goes back to Ajtai [Ajt96]. It states that a randomly chosen instance of a certain lattice problem is at least as hard as the worst-case instance of a related lattice problem. Thus, choosing secure keys is easy. This reduction was later on adapted to work with ideal lattices by Lyubashevsky and Micciancio [LM06] because ideal lattices offer a compact public-key representation and very efficient operations at the expense of a slightly stronger assumption.

The security model, mainly influenced by Juels, Luby, and Ostrovsky [JLO97] as well as Pointcheval and Stern [PS00], requires blind signature schemes to satisfy blindness and one-more unforgeability. Blindness states that the signer must not obtain any information on the signed messages and one-more unforgeability means that an adversary cannot obtain more signatures than there were interactions with the signer.

Our Contribution. We construct the first lattice-based blind signature scheme. It is inspired by Lyubashevsky's ID scheme [Lyu08a] in combination with the Fiat-Shamir paradigm [FS86]. It is unconditionally blind, selective-failure blind [CNS07], and one-more unforgeable in the random oracle model [BR93] if standard lattice problems in ideal lattices [LM06] are hard in the worst-case. [1] Then, the worst-case assumption is weaker as it refers to all lattices, instead of all *ideal* lattices. However, the ideal lattice version is much more efficient. With its four moves it is quite efficient. All operations have quasi-linear complexity and all keys and signatures require a quasi-linear amount of storage bits, with respect to the main parameter n. Moreover, it is leakage resilient according to a model inspired by Katz and Vaikuntanathan [KV09]. Let L be the bit-length of the secret key. Our scheme remains secure, even if the adversary obtains $L(1 - o(1))$ bits of the secret key via arbitrary side channels. This brings the security

[1] Notice that our scheme can instantiated with (regular) q-ary lattices as well.

model closer to reality, where the adversary may obtain information about the secret key, e.g, via (remote) timing attacks or by having physical access to the signing device. When applied in e-voting or e-cash schemes, such a resilience also helps against insider attacks and may improve the trust that we are willing to grant these schemes. Another application of our construction is identity-based blind signatures, when combined with [Rüc10].

Our scheme is also the first leakage resilient blind signature scheme and our results in this respect are applicable to Lyubashevsky's ID and signature schemes [Lyu08a, Lyu09]. It may be possible to use an analogue of Pointcheval and Stern's approach [PS00] to turn the leakage resilient variants [KV09, ADW09] of the Okamoto-Schnorr signature scheme [Sch91, Oka92] into blind signature schemes. However, it is unclear whether this will actually work and whether it will be efficient.

Table 1 compares RSA and Okamoto-Schnorr (OS) blind signatures with our construction in terms of computational cost. For all schemes, we propose parameter sets for current, medium, and future security levels. We believe that RSA is a good basis for comparison because it is easy to understand and very efficient as signing only involves two modular exponentiations and verification can be done in a single one (small exponent). We do *not* count multiplications. As observed in [BNPS03], the security of the RSA blind signature scheme is based on a specially tailored interactive assumption that is stronger than the original RSA assumption [BMV08]. Taking all this into account, the timings observed for RSA provide an optimistic lower bound for current practical and secure schemes. The timings for OS are expected timings based on the number of modular exponentiations, *not* counting multiplications. We include OS because it follows the typical 3-move structure and is based on a standard assumption. It is therefore closer to our protocol. The timings were obtained on an AMD Opteron CPU, running at 2.3 GHz. For RSA and OS, we have used OpenSSL 0.9.8g, which is supposed to be very efficient. For our blind signature schemes, we did a straightforward implementation, which certainly leaves room for improvements. Here, the timings reflect the *full* scheme.

From Table 1, we clearly see that our scheme benefits from its quasi-linear complexity, especially in higher levels of security. In addition, for our scheme, we can have various trade-offs between signature size and speed. For more details, refer to Appendix C. There, we also show how to optimize the key and signature sizes, which are typically large in lattice-based constructions.

We believe that our work is an important contribution because the previous efficient constructions, such as [Cha82, PS97, PS00, Abe01, BNPS03, CKW04, Oka06], have one thing in common: they are built upon classic number theoretic assumptions, like the hardness of factoring large integers or computing discrete logarithms. The more recent approaches, e.g., by Boldyreva [Bol03] or Okamoto [Oka06], tend to use pairings that yield very elegant constructions. They, however, are again based on the discrete logarithm problem in this specific setting. None of the above schemes remains secure in the presence of reasonably large quantum computers, where both factoring and computing discrete logarithms become easy due to the seminal work of Shor [Sho97].

Finally, we would like to mention that there are also (typically inefficient) instantiations from general assumptions, e.g., by Juels et al. [JLO97], Fischlin [Fis06], Hazay,

Scheme	Secure until	Security (bits)	Moves	KeyGen	Sign	Verify
RSA-1229	2012	Current (76)	2	95 ms	16 ms	5 ms
RSA-3313	2050	Medium (102)	2	1250 ms	46 ms	6 ms
RSA-15424	2282	Future (256)	2	251849 ms	2134 ms	20 ms
OS-1229	2012	Current (76)	3	16 ms	64 ms	24 ms
OS-3313	2050	Medium (102)	3	46 ms	184 ms	69 ms
OS-15424	2282	Future (256)	3	2134 ms	8536 ms	3201 ms
Section 3 ($n = 1024$)	2012	Current (76)	4	37 ms	220 ms	33 ms
Section 3 ($n = 2048$)	2050	Medium (102)	4	52 ms	283 ms	57 ms
Section 3 ($n = 8192$)	2282	Future (256)	4	305 ms	1175 ms	320 ms

The table compares our scheme with RSA and Okamoto-Schnorr for various moduli according to [Len05] (Current, Medium) and [ECR10] (Future). The bitlengths can be computed on `www.keylength.com`. For our blind signature scheme, we propose three *optimized* parameter sets for the same security levels based on [RS10], which provides a framework for choosing secure parameters for lattice-based cryptography. Note that the parameters for RSA and OS do *not* take potential quantum-computer attacks into account. All timings are averaged over 1000 random instances.

Table 1: Comparison of RSA, Okamoto-Schnorr, and our blind signature scheme.

Katz, Koo, and Lindell [HKKL07], or Abe and Ohkubo [AO09]. Whether they are post-quantum or leakage-resilient depends on the exact instantiation.

Main Obstacles. For every blind signature scheme, one has to overcome three basic obstacles. The scheme needs to be blind, one-more unforgeable, and at the same time complete. Blindness and unforgeability are already somewhat orthogonal because granting the user too much power to ensure blindness harms unforgeability and vice-versa. Since working with lattices, we do not have access to a cyclic group structure as in schemes that are based on the DDH or DL assumptions. There, blindness is typically easier to achieve by multiplying the message with a random group element. The result is again a random group element.

In lattices, we need to emulate this over an infinite structure via a filtering technique that is inspired by [Lyu08a]. However, this technique introduces a completeness defect that even affects the interaction of an honest user with an honest signer. Thus, the protocol may need to be restarted. We show how this technique can be refined to allow a time-memory trade-off, reducing the number of expected restarts at the expense of only slightly larger signatures. When addressing this defect, we need additional means to ensure blindness over repetitions of the protocol. Our solution involves a statistically hiding commitment.

Similarly, the completeness defect has implications with respect to unforgeability as the user may claim that the protocol has failed, whereas it was indeed successful. Here, we extend the typical three-move structure to a four-move structure where the user needs to demonstrate that he or she could not obtain a valid signature. Such a last

move, from user to signer, is highly unusual for blind signature schemes. We solve this issue by designing a special proof of failure and by employing a computationally binding commitment scheme.

All these issues, and the additional leakage resilience, need to be addressed simultaneously as they are interconnected. This leads to an intricate process of correctly setting up the numerous parameters and sets for our scheme in Table 2.

RSA-style Blind Signatures. One might think that RSA-style (hash $\rightarrow$ blind $\rightarrow$ invert $\rightarrow$ unblind) lattice-based blind signatures can be implemented using the preimage sampleable trapdoor function $f : D \subset \mathbb{Z}^m \rightarrow \mathbb{Z}_q^n$ from [GPV08]. If certain lattice problems are hard, it is hard to sample preimages from D (small norm) unless one knows short vectors x such that $f(x) = 0$. The user would hash the message M using a full-domain hash $h \leftarrow \mathsf{H}(M)$ and blind using $M^* \leftarrow h + f(\beta)$ for $\beta \in D$. The signer would *sample* from $f^{-1}(M^*) \cap D$ and return the result σ^*. The function is compressing, so there are no unique preimages. Using β and the fact that f is linear, the user can compute $\sigma \leftarrow \sigma^* - \beta$, which passes verification: $f(\sigma) = f(\sigma^*) - f(\beta) = \mathsf{H}(M^*)$. For the proof, one would rely on an interactive "one-more" trapdoor inversion assumption akin to [BNPS03]. However, the adversary must never obtain a non-zero $x \in D$ such that $f(x) = 0$ because this would imply learning a piece of the secret key. Unfortunately, such an attack is easy: take $u \in D$ and send $M^* = f(u)$ to the signer, who returns σ^*. Now, $x = u - \sigma^*$ is small and $f(x) = 0$. Also, $x \neq 0$ with high probability because there are many preimages of $f(u)$.

Organization. After a brief preliminaries section, we propose our blind signature scheme in Section 3. There, we also provide a detailed analysis, including completeness, blindness, one-more unforgeability, and leakage resilience. In Appendix C, we discuss how to choose practical parameters and Appendix D contains all supporting lemmas for the theorems in Section 3.

2. Preliminaries

With n, we always denote the security parameter. The joint execution of two algorithms $\mathcal{A}$ and $\mathcal{B}$ in an interactive protocol with private inputs x to $\mathcal{A}$ and y to $\mathcal{B}$ is written as $(a, b) \leftarrow \langle \mathcal{A}(x), \mathcal{B}(y) \rangle$. The private outputs are a for $\mathcal{A}$ and b for $\mathcal{B}$. Accordingly, $\langle \mathcal{A}(x), \mathcal{B}(y) \rangle^k$ means that the interaction can take place up to k times. The statement $x \leftarrow_\$ X$ means that x is chosen uniformly at random from the finite set X. Recall that the statistical distance of two random variables X, Y over a discrete domain D is defined as $\Delta(X, Y) = 1/2 \sum_{a \in D} |\operatorname{Prob}[X = a] - \operatorname{Prob}[Y = a]|$. A function is negligible if it vanishes faster than $1/p(n)$ for any polynomial p. All logarithms are base 2 and we identify $\{1, \ldots, k\}$ with $[k]$.

We recall the definitions of blind signatures and commitments. Afterwards, we briefly recall some facts from lattice theory. For the reader's convenience, the Forking Lemma from [BN06] is repeated in Appendix A.

Experiment $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}}(n)$
$b \leftarrow_\$ \{0,1\}$
$(\mathsf{pk},\mathsf{sk}) \leftarrow_\$ \mathsf{BS.Kg}(1^n)$
$(M_0, M_1, \mathsf{state}_{\mathrm{find}}) \leftarrow_\$ \mathcal{S}^*(\mathrm{find}, \mathsf{sk}, \mathsf{pk})$
$\mathsf{state}_{\mathrm{issue}} \leftarrow_\$ \mathcal{S}^{*\langle\cdot,\mathcal{U}(\mathsf{pk},M_b)\rangle^1,\langle\cdot,\mathcal{U}(\mathsf{pk},M_{1-b})\rangle^1}(\mathrm{issue}, \mathsf{state}_{\mathrm{find}})$
Let $\mathbf{s}_b$ and $\mathbf{s}_{1-b}$ be the outputs of $\mathcal{U}(\mathsf{pk}, M_b)$
and $\mathcal{U}(\mathsf{pk}, M_{1-b})$, respectively.
If $\mathbf{s}_0 \neq \mathsf{fail}$ and $\mathbf{s}_1 \neq \mathsf{fail}$
  $d \leftarrow_\$ \mathcal{S}^*(\mathrm{guess}, \mathbf{s}_0, \mathbf{s}_1, \mathsf{state}_{\mathrm{issue}})$
Else
  $d \leftarrow_\$ \mathcal{S}^*(\mathrm{guess}, \mathsf{fail}, \mathsf{fail}, \mathsf{state}_{\mathrm{issue}})$
Return 1 iff $d = b$

Experiment $\mathsf{Exp}^{\mathsf{omf}}_{\mathcal{U}^*,\mathsf{BS}}(n)$
$\mathsf{H} \leftarrow_\$ \mathcal{H}(1^n)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow_\$ \mathsf{BS.Kg}(1^n)$
$\{(M_1,\mathbf{s}_1),\ldots,(M_\jmath,\mathbf{s}_\jmath)\} \leftarrow_\$ \mathcal{U}^{*\mathsf{H}(\cdot),\langle\mathcal{S}(\mathsf{sk}),\cdot\rangle^\infty}(\mathsf{pk})$
Let ℓ be the number of successful interaction
between $\mathcal{U}^*$ and the signer.
Return 1 iff
  1. $M_i \neq M_j$ for all $1 \leq i < j \leq \jmath$;
  2. $\mathsf{BS.Vf}(\mathsf{pk}, \mathbf{s}_i, M_i) = 1$ for all $i = 1,\ldots,\jmath$;
  3. $\ell + 1 = \jmath$.

Figure 1: Security experiments for blindness and one-more unforgeability of blind signatures.

2.1. Blind Signatures

A blind signature scheme BS consists of three algorithms $(\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$, where Sign is an interactive protocol between a signer $\mathcal{S}$ and a user $\mathcal{U}$. The specification is as follows.

Key Generation. $\mathsf{Kg}(1^n)$ outputs a private signing key sk and a public verification key pk.

Signature Protocol. $\mathsf{Sign}(\mathsf{sk}, M)$ describes the joint execution of $\mathcal{S}$ and $\mathcal{U}$. The private output of $\mathcal{S}$ is a view $\mathcal{V}$ and the private output of $\mathcal{U}$ is a signature $\mathbf{s}$ on the message $M \in \mathcal{M}$ with message space $\mathcal{M}$ under sk. Thus, we write $(\mathcal{V}, \mathbf{s}) \leftarrow \langle\mathcal{S}(\mathsf{sk}), \mathcal{U}(\mathsf{pk}, M)\rangle$.

Signature Verification. The algorithm $\mathsf{Vf}(\mathsf{pk}, \mathbf{s}, M)$ outputs 1 if $\mathbf{s}$ is a valid signature on M under pk and otherwise 0.

Completeness is defined as with digital signature schemes, i.e., every honestly created signature for honestly created keys and for any messages $M \in \mathcal{M}$ has to be valid under this key. Views are interpreted as random variables, whose output is generated by subsequent executions of the respective protocol. Two views $\mathcal{V}_1$ and $\mathcal{V}_2$ are considered equal if they cannot be distinguished by any computationally unbounded algorithm with noticeable probability.

As for security, blind signatures have to satisfy two properties: blindness and one-more unforgeability [JLO97, PS00]. The notion of blindness is defined in the experiment $\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}}$ in Figure 1, where the adversarial signer $\mathcal{S}^*$ works in three modes. In mode *find*, it chooses two messages M_0, M_1 and interacts with two users in mode *issue*. Depending on a coin flip b, the first (second) user obtains a blind signature for M_b (M_{1-b}). After seeing the unblinded signatures in the original order, with respect to M_0, M_1, the signer has to guess the bit b in mode *guess*. If either of the user algorithms fails in outputting a valid signature, the signer is merely notified of the failure and does not get any signature. Below, we deal with aborts as an extension. Also note that we allow the adversary to keep a state that is fed back in subsequent calls. A scheme BS is (t,δ)-blind, if there is no adversary $\mathcal{S}^*$, running in time at most t, that wins the above experiment with advantage at least δ, where the advantage is defined as $\mathsf{Adv}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}} = \left|\mathrm{Prob}\left[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{S}^*,\mathsf{BS}}(n) = 1\right] - \frac{1}{2}\right|$. A scheme is *statistically* blind if the it is

(∞, δ)-blind for a negligible δ. The second security property, one-more unforgeability, ensures that each completed interaction between signer and user yields at most one signature. It is formalized in the experiment $\mathsf{Exp}^{\mathsf{omf}}_{\mathcal{U}^*,\mathsf{BS}}$, where an adversarial user tries to output $\jmath$ valid signatures after $\ell < \jmath$ completed interactions with an honest signer. $\mathcal{H}$ is a family of random oracles.

A signature scheme BS is $(t, q_{\mathsf{Sign}}, q_{\mathsf{H}}, \delta)$-one-more unforgeable if there is no adversary $\mathcal{A}$, running in time at most t, making at most q_{Sign} signature queries and at most q_{H} hash oracle queries, that wins the above experiment with probability at least δ.

2.2. Extensions

We consider three extensions to the above security model for blind signatures: one deals with user aborts, the second with dishonestly chosen keys, and the third with leakage resilience.

Security Under Aborts. Blindness in the previous subsection does not cover the case where the protocol is aborted prematurely. There is the strengthened notion of selective failure blindness [CNS07], where the malicious signer may choose either M_0 or M_1 according to some secret distribution that makes the protocol fail. Preventing this generically is easy as was shown by Fischlin and Schröder in [FS09]. In the course of the discussion of our construction, we argue that it already is blind in this sense.

Adversely-chosen Keys. Consider the blindness experiment in [ANN06]. Instead of having the experiment select pk, sk, we can let the signer output pk. Blindness may be harder to achieve in this setting. However, our construction remains blind in this stronger model as the proof does not exploit specifics about the key.

Leakage Resilience. Resilience to key leakage is a way to ensure security against side-channel attacks. In [KV09], Katz and Vaikuntanathan give a nice overview of past developments and the evolution of leakage resilience for authenticity [ADW09, KV09] and secrecy, e.g., [DP08, AGV09, NS09]. Obviously, we are interested in authenticity in the special case of blind signatures. We model key leakage in the unforgeability experiment by adding a leakage oracle $\mathsf{Leak}(\cdot)$ to $\mathsf{Exp}^{\mathsf{omf}}_{\mathcal{U}^*,\mathsf{BS}}$. The adversary can adaptively query Leak with a series of functions f_i, $i \in \{1, \ldots, \kappa\}$, and receives $f_i(\mathsf{sk})$. The only restriction is that $\sum_{i=1}^{n} |f_i(\mathsf{sk})| < \lambda(|\mathsf{sk}|)$, where the function λ determines the amount of leakage that we are willing to tolerate. Notice that the signer's key does not have to evolve over time and its secret state consists of the secret key only. Furthermore, observe that this extension is only sensible as long as $\lambda(|\mathsf{sk}|) < \min\{|\mathsf{sk}|, |\mathbf{s}|\}$, where $|\cdot|$ denotes bit-length and $\mathbf{s}$ is a signature. Otherwise, the adversary could easily obtain the entire secret key or a signature of its choice. See Appendix B for the experiment. To demonstrate leakage resilience, one has to show that the conditional min-entropy $H_\infty(\mathsf{sk}|\mathsf{Leak}(\mathsf{sk})) = \min_{\mathsf{sk}'}\{-\log(\mathrm{Prob}\left[\mathsf{sk} = \mathsf{sk}'|\mathsf{Leak}(\mathsf{sk})\right])\}$ of the secret key is still sufficiently large to prove security.

2.3. Commitments

Commitments typically work in two phases. First, one party publishes a commitment $C = \mathsf{com}(M; r) \in \{0,1\}^n$, $r \leftarrow_{\$} \{0,1\}^n$, to a message $M \in \{0,1\}^*$ without revealing any information about it. This is the "hiding" property of the commitment scheme. In the second phase, the party can prove that C actually corresponds to M by revealing r. It is important that no algorithm can find a second message M' and randomness r' such that $C = \mathsf{com}(M'; r')$, i.e., break the "binding" property. As usual, these properties are defined for families of such commitment functions. A scheme is (t, δ)-hiding (-binding) if there is no algorithm running in time at most t that can break the hiding (binding) property with probability at least δ. Both properties can be satisfied computationally or unconditionally but there is no scheme that is unconditionally hiding *and* unconditionally binding [Gol04].

For our scheme, we assume a statistically $\delta^{(h)}_{\mathsf{com}}$-hiding and computationally $(t_{\mathsf{com}}, \delta^{(b)}_{\mathsf{com}})$-binding commitment scheme. As we are interested in fully lattice-based schemes, we would like to point out that such commitment schemes can be built upon hard lattice problems [KTX08] but in practice, one rather uses cryptographic hash functions as a message authentication code. For example, using a lattice-based hash function [ADL+08].

2.4. Lattices

A lattice in $\mathbb{R}^n$ is a discrete set $\Lambda = \{\sum_{i=1}^{d} x_i \, \mathbf{b}_i \,|\, x_i \in \mathbb{Z}\}$, where $\mathbf{b}_1, \ldots, \mathbf{b}_d$ are linearly independent over $\mathbb{R}$. The matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_d]$ is a basis of the lattice Λ and we write $\Lambda = \Lambda(\mathbf{B})$. The dimension of the lattice is d. The main computational problem in lattices is the shortest vector problem (SVP), where an algorithm is given a description, a basis, of a lattice Λ and is supposed to find the shortest vector $\mathbf{v} \in \Lambda \setminus \{\mathbf{0}\}$ with respect to a certain ℓ_p norm (up to an approximation factor). More precisely, find a vector $\mathbf{v} \in \Lambda \setminus \{\mathbf{0}\}$, such that $\|\mathbf{v}\|_p \leq \gamma \, \|\mathbf{w}\|_p$ for all $\mathbf{w} \in \Lambda \setminus \{\mathbf{0}\}$ for a fixed approximation factor $\gamma \geq 1$.

In this work, we are interested in a special family of lattices related to ideals in the ring $\mathbf{R} = \mathbb{Z}_q[X]/\langle \mathbf{g} \rangle$, where q is prime and $\mathbb{Z}_q = \{-(q-1)/2, \ldots, (q-1)/2\}$. We focus on $\mathbf{g} = X^n + 1$ and $n =$ "power of two" for efficiency reasons but it may be replaced by any irreducible polynomial over $\mathbb{Z}$. Then, our scheme and the analysis become only slightly more involved. We identify $\mathbf{f} \in \mathbf{R}$ with its coefficient vector $\mathbf{f} = (f_0, \ldots, f_{n-1}) \in \mathbb{Z}_q^n$. Furthermore, we denote elements of the $\mathbf{R}$-module $\mathbf{R}^m$ with $\hat{\mathbf{a}} = (\mathbf{a_0}, \ldots, \mathbf{a_{m-1}})$ or directly with $(a_0, \ldots, a_{mn-1}) \in \mathbb{Z}_q^{mn}$. Consequently, we define $\|\mathbf{f}\|_\infty = \|(f_0, \ldots, f_{n-1})\|_\infty$. The norm on $\mathbf{R}$ is a slight abuse of notation, but it will only be used if $\mathbf{f}$ has small coefficients over $\mathbb{Z}$. A lattice corresponds to an ideal $I \subset \mathbf{R}$ if and only if every lattice vector is the coefficient vector of a polynomial in I. The SVP problem easily translates to ideal lattices, where we call it ideal-SVP (ISVP).

The average-case hardness assumption for our construction relies on the problem of finding short vectors in the kernel of the family $\mathcal{H}(\mathbf{R}, m)$ of module homomorphisms $h_{\hat{\mathbf{a}} \in \mathbf{R}^m} : \mathbf{R}^m \to \mathbf{R}, \hat{\mathbf{x}} \mapsto \hat{\mathbf{a}} \circledast \hat{\mathbf{x}} = \sum_{i=0}^{m-1} \mathbf{a_i} \mathbf{x_i}$, when restricting the domain to $D' \subset \mathbf{R}$,

i.e., restricting the coefficients in the input to $[-2d, 2d] \cap \mathbb{Z}$. This problem can be stated as the following collision problem [LM06].

Definition 2.1 (Collision Problem) *The collision problem $Col(\mathcal{H}(\mathbf{R}, m), D)$ asks to find a distinct pair $(\hat{\mathbf{x}}, \hat{\mathbf{x}}') \in D^m \times D^m$ such that $h(\hat{\mathbf{x}}) = h(\hat{\mathbf{x}}')$ for $h \leftarrow_{\$} \mathcal{H}(\mathbf{R}, m)$.*

Obviously, the function is linear over $\mathbf{R}^m$, i.e., $h(\mathbf{a}(\hat{\mathbf{x}} + \hat{\mathbf{y}})) = \mathbf{a}(h(\hat{\mathbf{x}}) + h(\hat{\mathbf{y}}))$ for all $\mathbf{a} \in \mathbf{R}$, $\hat{\mathbf{x}}, \hat{\mathbf{y}} \in \mathbf{R}^m$. In addition, solving $Col(\mathcal{H}(\mathbf{R}, m), D)$ implies being able to solve ISVP^{∞} in *every* lattice that corresponds to an ideal in $\mathbf{R}$ by the following theorem.

Theorem 2.2 (Worst-case to Average-case, Theorem 2 in [LM06]) *Let $D = \{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_{\infty} \leq d\}$, $m > \log(q)/\log(2d)$, and $q \geq 4dmn\sqrt{n}\log(n)$. An adversary $\mathcal{C}$ that solves the $Col(h, D)$ problem, i.e., finds distinct preimages $\hat{\mathbf{x}}, \hat{\mathbf{y}} \in D^m$ such that $h(\hat{\mathbf{x}}) = h(\hat{\mathbf{y}})$, can be used to solve ISVP^{∞} with approximation factors $\gamma \geq 16dmn\log^2(n)$ in the worst case.*

3. Blind Signatures from Ideal Lattices

We construct a lattice-based blind signature scheme. It is secure in the random oracle model under a worst-case assumption in ideal lattices and its time *and* space complexity is quasi-optimal, $\widetilde{\mathcal{O}}(n)$.

The road map for this section is as follows: We describe the 4-move blind signature scheme BS. Then, we prove completeness, blindness, and one-more unforgeability. Proving completeness is non-trivial as we need to address an inevitable completeness defect. In the course of the discussion we show that it neither harms security nor efficiency. Afterwards, we prove that the scheme is statistically blind and that it is one-more unforgeable unless the collision problem $Col(\mathcal{H}(\mathbf{R}, m), D)$ is easy. In consequence, one-more unforgeability can be based on the worst-case hardness of the ISVP. After the main analysis, we prove that our scheme also supports leakage resilience.

Observe that the scheme requires lots of parameters that need to be carefully worked out. Their definition in Table 2 will be justified later in the analysis. We chose not to "unwind" the parameters d_s, d_ϵ, etc. because we need their relative size in the various lemmas below, making the proofs easier to understand. The asymptotics in the third column should help estimating their magnitude. The parameter d_ϵ is a constant 1 here but it can be increased if it is necessary to sign hash values of bit length $> n\log_2(3)$. The "usage" hint in the table points at the section, where they are most influential. As for selecting practical parameters, we refer the reader to Appendix C. There, we propose secure parameter sets based on the analysis in [RS10]. The appendix also includes a discussion on possible trade-offs for efficiency.

3.1. Informal Description

We give a detailed, slightly informal description of the protocol Steps 1-5 in Figure 2. For each step, we need a set of carefully chosen parameters from Table 2 to achieve

Parameter	Value	Asymptotics	Usage
n	power of 2	-	main security parameter
d_s D_s	positive integer constant $< q/(4n)$ $\{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_\infty \leq d_s\}$	$\mathcal{O}(1)$	secret key size, unforgeability set of secret keys
c_m	$> 1/\log(2d_s)$	$\widetilde{\mathcal{O}}(1)$	witness indistinguishability, leakage resilience
m	$\lfloor c_m \log(q) \rfloor + 1$	$\Omega(\log(n))$	worst-case to average-case reduction
D_ϵ	$\{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_\infty \leq 1 =: d_\epsilon\}$	$\mathcal{O}(1)$	hash output size
ϕ, ψ	positive integer constant ≥ 1	$\mathcal{O}(1)$	completeness, speed
D_α	$\{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_\infty \leq \psi n d_\epsilon =: d_\alpha\}$	$\mathcal{O}(n)$	blindness
D_{ϵ^*}	$\{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_\infty \leq d_\alpha - d_\epsilon =: d_{\epsilon^*}\}$	$\mathcal{O}(n)$	blindness
D_y	$\{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_\infty \leq \phi m n^2 d_s d_{\epsilon^*} =: d_y\}$	$\widetilde{\mathcal{O}}(n^3)$	witness indistinguishability
G_*	$\{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_\infty \leq d_y - n d_s d_{\epsilon^*} =: d_{G_*}\}$	$\widetilde{\mathcal{O}}(n^3)$	witness indistinguishability, completeness defect
D_β	$\{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_\infty \leq \phi m n d_{G_*} =: d_\beta\}$	$\widetilde{\mathcal{O}}(n^4)$	blindness
G	$\{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_\infty \leq d_\beta - d_{G_*} =: d_G\}$	$\widetilde{\mathcal{O}}(n^4)$	blindness, completeness defect
D	$\{\mathbf{f} \in \mathbf{R} : \|\mathbf{f}\|_\infty \leq d_{G_*} + d_\beta + n d_s d_\epsilon =: d_D\}$	$\widetilde{\mathcal{O}}(n^4)$	collisions under h
q	$\geq 4mn\sqrt{n}\log(n) d_D$, prime	$\widetilde{\Theta}(n^5\sqrt{n})$	worst-case to average-case reduction

The table defines all parameters and sets for our scheme. The sets are defined via a norm bound, for which we also state the asymptotic growth with respect to the security parameter n. The last column states the main usage for the individual parameter or set. Some sets introduce a completeness error to the scheme that can be reduced by increasing ϕ. Reducing this defect also significantly improves performance. All sets are subsets of the ring $\mathbf{R} = \mathbb{Z}_q[X]/(X^n + 1)$.

Table 2: Parameters for the security parameter n.

completeness and security. For each parameter, the table contains a hint at their main usage and justification in the analysis.

Basically, the protocol follows the structure of a 3-move identification scheme, which provides a witness-indistinguishable proof of knowledge. The signer proves knowledge of $\hat{\mathbf{s}} \in D_s^m$ such that $h(\hat{\mathbf{s}}) = \mathbf{S}$ with $\mathbf{S}$ being the public key.

We stick to this basic structure and let the signer transmit a commitment $\mathbf{Y} = h(\hat{\mathbf{y}})$ for a random value $\hat{\mathbf{y}} \in D_y^m$. The user computes a challenge ϵ^* as a function (involving H) of $\mathbf{Y}$ and the to-be-signed message M and sends it to the signer, which returns $\hat{\mathbf{z}}^* = \hat{\mathbf{s}}\epsilon^* + \hat{\mathbf{y}}$. Via the linearity of h, the user can verify that $h(\hat{\mathbf{z}}^*) = \mathbf{S}\epsilon^* + \mathbf{Y}$ using only public knowledge. Afterwards, the user has to transform the "blinded" signature $(\hat{\mathbf{z}}^*, \epsilon^*)$ into the regular signature $(\hat{\mathbf{z}}, \epsilon)$ for M.

However, to obtain a blind signature scheme from this strategy, we need to overcome three main obstacles. First, the scheme needs to be unforgeable, i.e., neither $\mathbf{Y}$ nor $\hat{\mathbf{z}}^*$ may leak too much information about the secret key. Second, ϵ^* and $\hat{\mathbf{z}}$ need to be distributed independently of the message M to ensure blindness. Third, we need to ensure completeness of the resulting protocol, which will be non-trivial because of the following filtering technique [Lyu08a, Lyu09].

Roughly speaking, we add two numbers $a \in [-A, A]$ and $b \leftarrow_\$ [-3A, 3A]$ and filter the output in the sense that we only reveal $c = a + b$ if $c \in [-2A, 2A]$. Otherwise, we choose a fresh b and try again. This ensures that c is distributed independently of a. However,

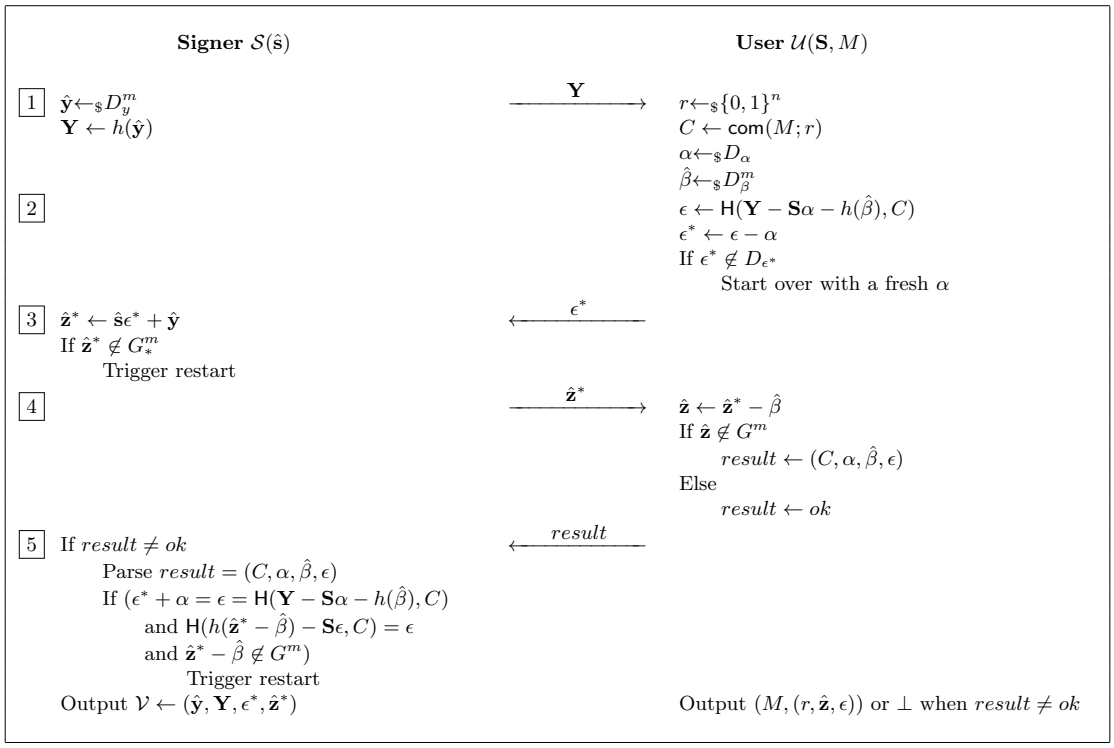

Figure 2: Issue protocol of the blind signature scheme BS. All parameters and sets are defined in Table 2. Note that the signer implicitly verifies that the user's protocol messages come from the correct domains.

the filtering technique in this example only works with probability $\approx 2/3$, but we expect $c \in [-2A, 2A]$ after $\approx 3/2$ trials.

It is applied to $(\hat{\mathbf{s}}\epsilon^*) + \hat{\mathbf{y}}$ to hide the secret key $\hat{\mathbf{s}}$ by randomly choosing the coefficients of $\hat{\mathbf{y}}$ from a relatively large set, compared to $\|\hat{\mathbf{s}}\epsilon^*\|_\infty$, and then filtering the output until it is in G_*^m. Whenever filtering fails in Step 3, the signer has to restart the entire protocol. After a small number of restarts, the signer can safely send $\hat{\mathbf{z}}^*$, bearing no information about the secret key. Actually, the filtering technique is much more involved and we need to deal with sums of vectors. We will show the details and a refinement in the next section.

Interestingly, the filtering technique can also be applied to achieve blindness. For the protocol message $\epsilon^* = \epsilon - \alpha$ after Step 2, we use $\alpha \leftarrow_{\$} D_\alpha^m$ to hide ϵ. This is necessary, as ϵ will be a part of the output signature. The completeness defect in this filtering step can be reduced to 0 because the user can repeat it locally. In Step 4, the user attempts to "unblind" the signature by computing $\hat{\mathbf{z}} \leftarrow \hat{\mathbf{z}}^* - \hat{\beta}$, where $\hat{\beta}$ is prepared in Step 1 and randomly chosen from a relatively large set to hide $\hat{\mathbf{z}}^*$. This is the third application of the filtering technique. If it fails, the protocol needs to be restarted again.

This last defect is the reason for having the last move and Step 5. Even if both parties

are honest, the user might not be able to obtain a valid signature with non-negligible probability (Step 4). This is highly unusual for a blind signature scheme, in fact we are not aware of any other schemes with this kind of behavior. In such a case, the user needs to prove that no valid signature could be obtained (Step 5) and the protocol needs to be restarted. Therefore, the user submits $(C, \alpha, \hat{\beta}, \epsilon)$ to the signer, where C is a commitment to the message M. The signer can verify that the user was unable to obtain a valid signature relative to the commitment C. More formally, we prove that successfully cheating in Step 5 implies being able to solve $Col(\mathcal{H}(\mathbf{R}, m), D)$. In addition, for unforgeability, we require that the commitment is binding and for blindness it is crucial that it is hiding. The hiding property of the commitment also prevents the signer from learning information about M across the required restarts.

Since restarts do not harm security, we can repeat the protocol until it is complete. The expected number of repetitions is constant and it can be brought very close to 1 by choosing the parameters appropriately.

3.2. Our Construction

We construct our blind signature scheme $\mathsf{BS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ as follows.

Key Generation. $\mathsf{BS.Kg}(1^n)$ selects a secret key $\hat{\mathbf{s}} \leftarrow_\$ D_s^m$, and a compression function $h \leftarrow_\$ \mathcal{H}(\mathbf{R}, m)$. Let $\mathcal{C}(1^n)$ be a commitment scheme, mapping $\{0,1\}^* \times \{0,1\}^n \to \{0,1\}^n$. The algorithm chooses a function $\mathsf{com} \leftarrow_\$ \mathcal{C}(1^n)$ and, in addition, selects $\mathsf{H} \leftarrow_\$ \mathcal{H}(1^n)$ mapping $\{0,1\}^* \to D_\epsilon \subset D$.
Then, it computes the public key $\mathbf{S} \leftarrow h(\hat{\mathbf{s}})$ and outputs $(\hat{\mathbf{s}}, \mathbf{S})$. For simplicity, we treat h, com, H, and the parameters in Table 2 as globally known and implicit inputs to all algorithms. However, each signer may choose them individually and include them in the public key.

Signature Protocol. The signature issue protocol for messages $M \in \{0,1\}^*$ is depicted in Figure 2. Eventually, the user outputs a message M and a signature $(r, \hat{\mathbf{z}}, \epsilon)$.
Notes: Upon a restart after Step 2, the user only selects a fresh $\alpha \leftarrow_\$ D_\alpha$ and repeats the operations that involve α. Whenever the signer triggers a restart, the user chooses a fresh r in order to make the protocol execution independent of the previous ones. Therefore, we omit values from previous runs in the signer's view. During Step 5, the signer can detect a cheating user that tries to trigger a restart, despite having received a valid signature. In this case, the signer can stop the protocol and assume that the user has obtained a valid signature.

Verification. $\mathsf{BS.Vf}(\mathbf{S}, (r, \hat{\mathbf{z}}, \epsilon), M)$ outputs 1 iff $\hat{\mathbf{z}} \in G^m$ and $\mathsf{H}(h(\hat{\mathbf{z}}) - \mathbf{S}\epsilon, \mathsf{com}(M; r)) = \epsilon$.

3.3. Analysis and Security

In this section, we analyze our blind signature scheme with regard to completeness, blindness, one-more unforgeability, and leakage resilience. For each aspect, we prove a main theorem. Supporting lemmas are stated before the theorems and proven in Appendix D.

3.3.1. Completeness

Completeness of BS is a non-trivial issue due to the eventual restarts and the many parameters involved. The next lemma ensures that the number of restarts is small, effectively constant.

Lemma 3.1 *Let* $k = \Omega(n)$, $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^k$ *with* arbitrary $\mathbf{a} \in \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \leq A\}$ *and* random $\mathbf{b} \leftarrow_\$ \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \leq B\}$. *Given* $B \geq \phi k A$ *for* $\phi \in \mathbb{N}_{>0}$, *we have* $\Prob_{\mathbf{b}}[\,\|\mathbf{a} - \mathbf{b}\|_\infty \leq B - A] > \frac{1}{e^{1/\phi}} - o(1)$.

The multiplication of two polynomials modulo X^n+1 plays a major role in the analysis. Therefore, we need the following lemma, which is a special case of [Lyu08b, Lemma 2.8].

Lemma 3.2 *For any two polynomials* $\mathbf{a}, \mathbf{b} \in \mathbf{R}$, *we have* $\|\mathbf{ab} \bmod (X^n + 1)\|_\infty \leq n \|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty$.

Notice that it is possible to prove a better bound $\sqrt{n}\log(n) \|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty$ that holds with overwhelming probability $p(n) = 1 - 4ne^{-\log^2(n)/8}$, but $p(n)$ does not converge fast enough to be of practical use.

Theorem 3.3 (Completeness) *Let* $g(n) = \omega(\log^2(n))$. *The scheme* BS *is complete after at most* $g(n)$ *(or, an expected number of* $e^{2/\phi}$*) repetitions.*

Proof. Let us assume that the protocol does not have to be restarted after Steps 2, 3, and 4. Then, for all honestly generated key pairs $(\hat{\mathbf{s}}, \mathbf{S})$, all messages $M \in \{0,1\}^*$, and all signatures $(r, \hat{\mathbf{z}}, \epsilon)$ we have $\hat{\mathbf{z}} \in G^m$ and $h(\hat{\mathbf{z}}) - \mathbf{S}\epsilon = h(\hat{\mathbf{z}}^* - \hat{\beta}) - \mathbf{S}\epsilon = h(\hat{\mathbf{s}}(\epsilon - \alpha) + \hat{\mathbf{y}} - \hat{\beta}) - \mathbf{S}\epsilon = \mathbf{Y} - \mathbf{S}\alpha - h(\hat{\beta})$ and $\mathsf{com}(M; r) = C$. Therefore $\mathsf{H}(h(\hat{\mathbf{z}}) - \mathbf{S}\epsilon, \mathsf{com}(M; r)) = \epsilon$ and $\mathsf{BS.Vf}(\mathbf{S}, (r, \hat{\mathbf{z}}, \epsilon), M) = 1$.

Now, we analyze the probability of a restart. Observe that the restarts after Step 2 do not affect completeness, as the user does them locally. The number of trials here is at most $g(n)$ for any $g(n) = \omega(\log(n))$ due to Lemma 3.1 ($k = n, A = d_s, B = d_\alpha$) for $\epsilon - \alpha \in D_{\epsilon^*}$. However, the expected number of trials is constant ($e^{1/\psi}$). It is safe to set $\psi = 1$ here but one might want less trials, e.g., less than 1.5 for $\psi \geq 3$ and $n > 1$.

After Steps 3 and 4, aborts affect the protocol and trigger a full restart. In Step 3, we need to ensure that $\hat{\mathbf{s}}\epsilon^* + \hat{\mathbf{y}} \in G_*^m$. By Lemma 3.2, we know that $\|\hat{\mathbf{s}}\epsilon^* \bmod (X^n + 1)\|_\infty \leq nd_s d_{\epsilon^*}$ and applying Lemma 3.1 ($k = mn, A = nd_s d_{\epsilon^*}, B = d_y$) yields the constant success probability $e^{-1/\phi}$ and a maximum number of trials of $\omega(\log(n))$. This can be optimized by increasing ϕ. After an expected number $e^{1/\phi}$, the protocol proceeds to Step 4.

In Step 4, the user attempts to "unblind" the signature and requires that $\hat{\mathbf{z}}^* - \hat{\beta} \in G^m$. Otherwise, the user convinces the signer that a restart is necessary. We apply Lemma 3.1 on ($k = mn, A = d_{G_*}, B = d_\beta$) and obtain the same behavior as in Step 3.

In total, after at most $g(n)$, for any $g(n) = \omega(\log^2(n))$, or an expected number $e^{2/\phi}$ of trials, the protocol is complete. □

In Appendix C, we will see that choosing $\phi = 4$ is good choice to make the protocol more efficient in practice. Observe that in any case, all operations (including eventual restarts) in BS have $\widetilde{\mathcal{O}}(n)$ complexity and that private keys, public keys, and signatures have size $\widetilde{\mathcal{O}}(n)$.

3.3.2. Blindness

We prove that BS is statistically blind based on the observation that the signer only sees values that are independent of the message being signed. More precisely, the views generated by two different messages are indistinguishable. For this argument to work, we require a statistically hiding commitment scheme and carefully selected sets D_α, D_β, D_{ϵ^*}, and G. The following probabilistic lemma is crucial as it guarantees that the user's message after Step 2 and the final output are independent of the message. In the context of $\mathsf{Exp}^{\mathsf{blind}}_{S^*,\mathsf{BS}}$, this establishes a form of witness indistinguishability w.r.t. the messages that are chosen by the malicious signer.

Lemma 3.4 *Let* $k \in \mathbb{N}$, $\mathbf{a}, \mathbf{a}', \mathbf{b} \in \mathbb{Z}^k$ *with* arbitrary $\mathbf{a}, \mathbf{a}' \in \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \leq A\}$, *a* random $\mathbf{b} \leftarrow_\$ \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \leq B\}$ *for* $B > A$. *We define the random variables* $\mathbf{c} \leftarrow \mathbf{a} - \mathbf{b}$ *and* $\mathbf{c}' \leftarrow \mathbf{a}' - \mathbf{b}$ *if* $\max\{\|\mathbf{a} - \mathbf{b}\|_\infty, \|\mathbf{a}' - \mathbf{b}\|_\infty\} \leq B - A$, *otherwise, we resample* $\mathbf{b}$. *Then,* $\Delta(\mathbf{c}, \mathbf{c}') = 0$.

The role of com is to ensure that the signer can only obtain negligible information from restarts. Notice that BS is perfectly blind ($(\infty, 0)$-blind) if the commitment scheme is perfect (0-hiding).

Theorem 3.5 (Blindness) BS *is* $(\infty, \delta^{(h)}_{\mathsf{com}})$*-blind if* com *is* $\delta^{(h)}_{\mathsf{com}}$*- hiding.*

Proof. As per experiment $\mathsf{Exp}^{\mathsf{blind}}_{S^*,\mathsf{BS}}$, the adversarial signer outputs two messages M_0, M_1 and interacts with two users $\mathcal{U}(\mathbf{S}, M_b)$, $\mathcal{U}(\mathbf{S}, M_{1-b})$ after a secret coin flip $b \leftarrow \{0, 1\}$. We show that these users do not leak any information about their respective message.

Technically, we establish that all protocol messages and the output, when interpreted as random variables, are distributed independently of the message being signed. This involves an analysis of ϵ^*, $\hat{\mathbf{z}}$, and eventual restarts. As for ϵ and r we need not worry. They are chosen uniformly at random.

Distribution of ϵ^*. Let $\epsilon^*_b, \epsilon^*_{1-b}$ be the first protocol messages of $\mathcal{U}(\mathsf{pk}, M_b)$ resp. $\mathcal{U}(\mathsf{pk}, M_{1-b})$. They are in D_{ϵ^*} and they are both of the form $\epsilon - \alpha$ with $\epsilon \in D_\epsilon$ and $\alpha \leftarrow_\$ D_\alpha$. The statistical distance $\Delta(\epsilon^*_b, \epsilon^*_{1-b})$ is 0 by Lemma 3.4 ($k = n, A = d_s, B = d_\alpha$) because the coefficients in D_{ϵ^*} are bounded by $B - A = d_\alpha - d_s$.

Distribution of $\hat{\mathbf{z}}$. Let $\hat{\mathbf{z}}_0, \hat{\mathbf{z}}_1$ be part of the final output of $\mathcal{U}(\mathsf{pk}, M_0)$ resp. $\mathcal{U}(\mathsf{pk}, M_1)$. Both are of the form $\hat{\mathbf{z}}^* - \hat{\beta}$ for $\hat{\mathbf{z}}^* \in G^m_*$ and $\hat{\beta} \leftarrow_\$ D^m_\beta$. Furthermore, $\hat{\mathbf{z}}_0$ and $\hat{\mathbf{z}}_1$ are forced to be in G^m, having coefficients bounded by $d_\beta - d_{G_*}$. Hence, the statistical distance $\Delta(\hat{\mathbf{z}}_0, \hat{\mathbf{z}}_1)$ is 0 because of Lemma 3.4 ($k = mn, A = d_{G_*}, B = d_\beta$).

Restarts. Observe that each protocol run is statistically independent of the previous runs by the statistical hiding property of the commitment com and because the user selects fresh $r, \alpha, \hat{\beta}$ after every restart. This is the reason why we inherit the

statistical $\delta^{(h)}_{\mathsf{com}}$-hiding property to obtain $(\infty, \delta^{(h)}_{\mathsf{com}})$-blindness instead of perfect blindness. Finally, we need to argue about the restart after Step 4. The user sends $(C, \alpha, \hat{\beta}, \epsilon)$ to the signer. These information allow the verification of the signature with respect to C. The message is still statistically hidden by the hiding property of com because the user never reveals the decommitment r.

Hence, the protocol hides the to-be-signed message and subsequent runs of the protocol for the same message are statistically independent. □

Furthermore, our scheme already supports selective failure blindness as shown in [FS09] because we are signing commitments instead of the adversely chosen messages. Even the fourth move does not reveal any information about the message due to the hiding property of the commitment.

3.3.3. One-more Unforgeability

In this section, we show that BS is one-more unforgeable, provided that the collision problem $Col(\mathcal{H}(\mathbf{R}, m), D)$ is hard and the commitment scheme is binding. The main tool in the reduction is the Forking Lemma [PS00, BN06]. To simulate the environment, especially blind signature queries, for the attacker $\mathcal{A}$ in the unforgeability experiment, we require that there are at least two possible secret keys for each public key $\mathbf{S}$ (Lemma 3.6). Moreover, we need the signature protocol to be witness indistinguishable to prevent the attacker from learning the secret key (Lemma 3.7). The binding property of com is necessary to prevent an attacker from obtaining one signature that works for two messages by changing the message under the commitment. All other attackers output at least one signature that does not correspond to a completed interaction. Here, we apply the Forking Lemma to extract knowledge about the secret key that was used to compute the forgery. Using this knowledge the reduction can solve the collision problem. Finally, we need to deal with Step 5 in the protocol. The adversary proves that it was unable to obtain a valid signature. We show that this is sufficient if Col is hard.

Since the function family $\mathcal{H}(\mathbf{R}, m)$ compresses the domain D_s^m, it is easy to show that all secret keys collide with at least one other secret key.

Lemma 3.6 *Let $h \in \mathcal{H}(\mathbf{R}, m)$. For every secret key $\hat{\mathbf{s}} \leftarrow_{\$} D_s^m$, there is a second $\hat{\mathbf{s}}' \in D_s^m \setminus \{\hat{\mathbf{s}}\}$ with $h(\hat{\mathbf{s}}) = h(\hat{\mathbf{s}}')$ (with overwhelming probability).*

The next lemma establishes witness indistinguishability of the protocol. Witness indistinguishability ensures that the malicious verifier cannot distinguish whether the prover uses one of two possible secret keys $\hat{\mathbf{s}}, \hat{\mathbf{s}}' \in h^{-1}(\mathbf{S}) \cap D_s^m$. Basically, it can be interpreted as an application of Lemma 3.4 to $\hat{\mathbf{z}}^* = (\hat{\mathbf{s}}\epsilon^*) + \hat{\mathbf{y}} \in G_*^m$ with some further observations. The choice of $\hat{\mathbf{y}} \leftarrow_{\$} D_y$ and the restriction "$\in G_*^m$" hide the first summand.

Lemma 3.7 *Let $h \in \mathcal{H}(\mathbf{R}, m)$ and $\mathbf{S} \in \mathbf{R}$. For any message M and any two secret keys $\hat{\mathbf{s}}, \hat{\mathbf{s}}' \in D_s^m$ with $h(\hat{\mathbf{s}}) = \mathbf{S} = h(\hat{\mathbf{s}}')$, the resulting protocol views $(\mathbf{Y}, \epsilon^*, \hat{\mathbf{z}}^*)$ and $(\mathbf{Y}', \epsilon^{*\prime}, \hat{\mathbf{z}}^{*\prime})$ are indistinguishable.*

Using lemmas 3.6 and 3.7, we can exploit witness indistinguishability to simulate all blind signature oracle queries with a secret key $\hat{\mathbf{s}}$ and at the same time expect the adversary to output a forgery that corresponds to a different secret key $\hat{\mathbf{s}}'$ with non-negligible probability or break the binding property of the commitment scheme. We apply the Forking Lemma to extract a solution to the $Col(\mathcal{H}(\mathbf{R}, m), D)$.

Theorem 3.8 (One-more unforgeability) *Let* Sig *be the signature oracle. Let* T_{Sig} *and* T_{H} *be the cost functions for simulating the oracles* Sig *and* H*, and let* $c < 1$ *be the probability for a restart in the protocol.* BS *is* $(t, q_{\mathsf{Sign}}, q_{\mathsf{H}}, \delta)$*-one-more unforgeable if* com *is* $(t', \delta/2)$*-binding and* $Col(\mathcal{H}(\mathbf{R}, m), D)$ *is* $(t', \delta'/2)$*-hard with* $t' = t + q_{\mathsf{H}}^{q_{\mathsf{Sig}}}(q_{\mathsf{Sign}} T_{\mathsf{Sig}} + q_{\mathsf{H}} T_{\mathsf{H}})$ *and non-negligible* δ' *if* δ *is non-negligible.*

The probability δ' depends on the number of issued signatures. It can be found at the end of the proof.

Proof. Towards contradiction, we assume that there exists a successful forger $\mathcal{A}$ against one-more unforgeability of BS with non-negligible probability δ. Using $\mathcal{A}$, we construct an algorithm $\mathcal{B}$, such that it either solves the collision problem or breaks the binding property of com.

Setup. $\mathcal{B}$ flips a coin $b \leftarrow_{\$} \{0,1\}$. For $b = 0$, it selects $h \leftarrow_{\$} \mathcal{H}(\mathbf{R}, m)$. For $b = 1$, it gets the description of h as input. $\mathcal{B}$ initializes a list $L_{\mathsf{H}} \leftarrow \emptyset$ of query-hash pairs $(\mathbf{R} \times \{0,1\}^*, D_\epsilon)$. It chooses $\hat{\mathbf{s}} \leftarrow_{\$} D_s^m$ and sets $\mathbf{S} \leftarrow h(\hat{\mathbf{s}})$. Furthermore, it randomly pre-selects random oracle answers $\mathbf{h}_1, \ldots, \mathbf{h}_{q_{\mathsf{H}}} \leftarrow_{\$} D_\epsilon$ and a random tape ρ. It runs $\mathcal{A}(\mathbf{S}; \rho)$ in a black-box simulation.

Random Oracle Queries. On input $(\mathbf{u}, C)$, $\mathcal{B}$ looks up $(\mathbf{u}, C)$ in L_{H}. If it finds corresponding hash value ϵ then it returns ϵ. Otherwise, $\mathcal{B}$ selects the first unused ϵ from the list $\mathbf{h}_1, \ldots, \mathbf{h}_{q_{\mathsf{H}}}$, stores $((\mathbf{u}, C), \epsilon)$ in L_{H}, and returns ϵ.

Blind Signature Queries. $\mathcal{B}$ acts according to the protocol in Figure 2.

Output. Eventually, $\mathcal{A}$ stops and outputs $(M_1, (r_1, \hat{\mathbf{z}}_1, \epsilon_1)), \ldots, (M_\jmath, (r_\jmath, \hat{\mathbf{z}}_\jmath, \epsilon_\jmath))$, $q_{\mathsf{Sign}} + 1 = \jmath$, for distinct messages. If $b = 0$, the reduction looks for two pairs $(M_1^*, (r_1^*, \hat{\mathbf{z}}^*, \epsilon^*))$ and $(M_2^* \neq M_1^*, (r_2^*, \hat{\mathbf{z}}^*, \epsilon^*))$ and outputs $(M_1^*, r_1^*), (M_2^*, r_2^*)$ to break the binding property of com. If there is no such collision, $\mathcal{B}$ aborts. If $b = 1$, the simulator $\mathcal{B}$ guesses an index $k \leftarrow_{\$} [\jmath]$ such that $h_\imath = \epsilon_k$ for some $\imath \in [q_{\mathsf{H}}]$. Then, $\mathcal{B}$ starts over, running $\mathcal{A}(\mathbf{S}; \rho)$ with random oracle answers $\mathbf{h}_1, \ldots, \mathbf{h}_{\imath-1}, \mathbf{h}'_\imath, \ldots, \mathbf{h}'_{q_{\mathsf{H}}}$ for a fresh set $\mathbf{h}'_\imath, \ldots, \mathbf{h}_{q_{\mathsf{H}}} \leftarrow_{\$} D_\epsilon$. Both $\mathcal{A}$ and $\mathcal{B}$ are run with the same random tape as in the first run. Among other values, $\mathcal{A}$ outputs $(M'_k, (r'_k, \hat{\mathbf{z}}'_k, \epsilon'_k))$ and $\mathcal{B}$ returns $(\hat{\mathbf{z}}_k - \hat{\mathbf{s}}\epsilon_k, \hat{\mathbf{z}}'_k - \hat{\mathbf{s}}\epsilon'_k)$ if $\epsilon'_k = \epsilon_k$ in an attempt to solve $Col(\mathcal{H}(\mathbf{R}, m), D)$. If $\epsilon'_k \neq \epsilon_k$, the reduction retries at most $q_{\mathsf{H}}^\jmath$ times with a different random tape and random oracle.

Analysis. $\mathcal{A}$'s environment is perfectly simulated. Especially, restarts happen with the same probability as in the original protocol. For $b = 0$, $\mathcal{B}$ $(t', \delta/2)$-breaks the binding property of com if $\mathcal{A}$ breaks the binding property of com to break one-more unforgeability.

For $b = 1$, we assume that $\mathcal{A}$ breaks one-more unforgeability without attacking com. So, at least one of the output signatures is not obtained via an interaction. The probability that $\mathcal{B}$ guesses the index k of this signature correctly is at least $1/(q_{\mathsf{Sign}} + 1)$.

Observe that ϵ_k is a random oracle answer but with probability $1/|D_\epsilon|$. Furthermore, notice that with probability $1/2$, at least one of the re-runs of $\mathcal{A}$ yields the same map $\{(\imath, k) : h_\imath = \epsilon_k\}$ as in the first run of $\mathcal{A}$. Thus, we consider the indices in both "interesting" replays to be constant.

Applying the Forking Lemma, we know that with probability $\delta_{\mathsf{frk}} \geq (1-c)(\delta - 1/|D_\epsilon|)((\delta - 1/|D_\epsilon|)/q_{\mathsf{H}} - 1/|D_\epsilon|)$, $\mathcal{A}$ is again successful in the one-more unforgeability experiment and outputs $(M'_k, (r'_k, \hat{\mathbf{z}}'_k, \epsilon'_k))$ using the *same random oracle query* as in the first run. The additional $(1-c)$ factor takes a potential abort during the second run into account, which happen with probability at most c. Therefore, we know that $(h(\hat{\mathbf{z}}_k - \mathbf{S}\epsilon_k), \mathsf{com}(M_k; r_k)) = (h(\hat{\mathbf{z}}'_k - \mathbf{S}\epsilon'_k), \mathsf{com}(M'_k; r'_k))$.

Now, we turn to solving the collision problem. We have to show that $\hat{\mathbf{z}}_k - \hat{\mathbf{s}}\epsilon_k \neq \hat{\mathbf{z}}'_k - \hat{\mathbf{s}}\epsilon'_k$ and $h(\hat{\mathbf{z}}_k - \hat{\mathbf{s}}\epsilon_k) = h(\hat{\mathbf{z}}'_k - \hat{\mathbf{s}}\epsilon'_k)$. The second requirement follows directly from the previous paragraph. The first is more involved. Here, it is important that the protocol is witness indistinguishable (Lemma 3.7), i.e., the adversary does not recognize whether we have used one of at least two possible $\hat{\mathbf{s}}, \hat{\mathbf{s}}'$ (Lemma 3.6 with probability greater than $1/2$. Thus, with probability at least $1/2$ its output corresponds to $\hat{\mathbf{s}}'$. Furthermore, the index maps of output ϵ_i's and random oracle answers $\mathbf{h}_j$'s are constant in both runs. Both conditions allow us to apply [PS00, Lemma 8]. It states that the random variables $\chi_k = \hat{\mathbf{z}}_k - \hat{\mathbf{s}}\epsilon_k$ and $\chi'_k = \hat{\mathbf{z}}'_k - \hat{\mathbf{s}}\epsilon'_k$ will be sensitive to the modified random oracle answers for indices $\geq \imath$. Hence, $\chi_k \neq \chi'_k$ with probability at least $1/2$ and we obtain the desired collision with norm at most $d_G + nd_sd_\epsilon < d_D$. Otherwise, we would have $\hat{\mathbf{z}}_k - \hat{\mathbf{s}}\epsilon_k = \hat{\mathbf{z}}'_k - \hat{\mathbf{s}}\epsilon'_k$ and $\hat{\mathbf{z}}_k - \hat{\mathbf{s}}'\epsilon_k = \hat{\mathbf{z}}'_k - \hat{\mathbf{s}}'\epsilon'_k$. We subtract the equations and obtain $(\epsilon_k - \epsilon'_k)(\hat{\mathbf{s}}' - \hat{\mathbf{s}}) = \mathbf{0}$. We know that $\epsilon_k - \epsilon'_k \neq \mathbf{0}$. Now, $\|(\epsilon_k - \epsilon'_k)(\hat{\mathbf{s}}' - \hat{\mathbf{s}})\|_\infty \leq 4d_sn < q/2$ because $\|\epsilon_k - \epsilon'_k\|_\infty \leq 2$ and $\|\hat{\mathbf{s}}' - \hat{\mathbf{s}}\|_\infty \leq 2d_s$. Thus, $(\epsilon_k - \epsilon'_k)(\hat{\mathbf{s}}' - \hat{\mathbf{s}}) = \mathbf{0}$ over $\mathbb{Z}[X]/\langle X^n + 1\rangle$, which is an integral domain. So, we have the contradiction $\hat{\mathbf{s}}' = \hat{\mathbf{s}}$ and a collision $(\hat{\mathbf{z}}_k - \hat{\mathbf{s}}\epsilon_k, \hat{\mathbf{z}}'_k - \hat{\mathbf{s}}\epsilon'_k) \in D \times D$. The success probability is at least $\delta_{\mathsf{col}} \geq 1/4\,\delta_{\mathsf{frk}}/(q_{\mathsf{Sign}} + 1)$, which is non-negligible if δ is non-negligible.

Concerning restarts, we argue that the user cannot obtain a valid signature out of an aborted interaction without solving the collision problem. In order to trigger an abort after Step 4, it outputs $result = (C, \alpha, \hat{\beta}, \epsilon)$ which, together with $\hat{\mathbf{z}}^*, \hat{\mathbf{y}}, \epsilon^*$, satisfies all abort criteria:

$$\epsilon^* + \alpha = \epsilon \;=\; \mathsf{H}(\mathbf{Y} - \mathbf{S}\alpha - h(\hat{\beta}), C) \tag{1}$$

$$\epsilon \;=\; \mathsf{H}(h(\hat{\mathbf{z}}^* - \hat{\beta}) - \mathbf{S}\epsilon, C) \tag{2}$$

$$\hat{\mathbf{z}}^* - \hat{\beta} \;\notin\; G^m \tag{3}$$

Assume that it also obtains a valid signature $(r', \hat{\mathbf{z}}', \epsilon')$ from this interaction. If $\epsilon = \epsilon'$, then $h(\hat{\mathbf{z}}^* - \hat{\beta}^* - \hat{\mathbf{s}}\epsilon) = h(\hat{\mathbf{z}}' - \hat{\mathbf{s}}\epsilon)$ by (2). If the arguments under h are equal, we have $\hat{\mathbf{z}}^* - \hat{\beta} \in G^m$ — a contradiction with (3). If the arguments are distinct, we have a collision in D because $\|\hat{\mathbf{z}}' - \hat{\mathbf{s}}\epsilon\|_\infty \leq d_G < d_D$ and $\left\|\hat{\mathbf{z}}^* - \hat{\beta}^* - \hat{\mathbf{s}}\epsilon\right\|_\infty \leq d_{G_*} + d_\beta + nd_sd_\epsilon = d_D$.

The adversary may succeed by hiding $\epsilon' \neq \epsilon$ in ϵ^*. But then, we necessarily have $\epsilon^* = \epsilon - \alpha = \epsilon' - \alpha'$ by (1) for an $\alpha \neq \alpha'$ and we know that $\alpha = \epsilon - \epsilon' + \alpha'$. So, the adversary had to be able to predict the output of H to compute α.

To conclude, the probability that we can extract a collision from a cheating user during an abort is at least $\delta_{\mathsf{abort}} \geq \delta\,(1 - 1/|D_\epsilon|)$, which is non-negligible if δ is non-negligible. Thus, the overall success probability of the reduction is $\delta' \geq \min(\delta_{\mathsf{col}}, \delta_{\mathsf{abort}})$ if the guess $b = 1$ was correct. □

Hence, we require that $q_{\mathsf{Sig}} = o(n)$ to be able to rely on the subexponential hardness of lattice problems. This constraint is an artifact of the proof technique as discussed in [PS00] and it is not at all unusual for efficient blind signature schemes. There, it was even required that $q_{\mathsf{Sig}} \leq (\log(n))^{\mathcal{O}(1)}$ because they needed a polynomial-time reduction. In consequence, in our reduction, we greatly benefit from the subexponential hardness of the underlying lattice problem. Alternatively, we believe that the running time of the reduction can be significantly reduced to being polynomial in q_{Sig} by using techniques due to Pointcheval [Poi98].

By Theorem 2.2, we get the following strong worst-case security guarantees.

Corollary 3.9 BS *is one-more unforgeable if solving* ISVP^∞ *is hard in the worst case for approximation factors* $\gamma \geq 16 d_D mn \log^2(n) = \widetilde{\mathcal{O}}(n^5)$ *in lattices that correspond to ideals in* $\mathbf{R}$.

3.3.4. Leakage Resilience

Using an additional restriction for one of the parameters, we can safely leak a $(1 - o(1))$ fraction of the secret key in the unforgeability experiment according to the definition in Appendix B. Recall that $m = \lfloor c_m \log(q) \rfloor + 1$ for some $c_m = \widetilde{\mathcal{O}}(1)$. Thus, it is possible to choose c_m, say $\log(n)$, without loosing the scheme's quasi-optimal efficiency. The following theorem states that such a choice is sufficient to provide strong leakage resilience.

To prove the theorem, we use a technical lemma from [KV09] in its alternative interpretation.

Lemma 3.10 ([KV09, Lemma 1]) *Let X be a random variable with $H := H_\infty(X)$, and fix $H' \in [0, H]$. Let f be a function whose range has size 2^λ, and set $Y := \{y \in \{0,1\}^\lambda | H_\infty(X|y = f(X)) \geq H'\}$. Then* $\mathrm{Prob}[\, f(X) \in Y] \geq 1 - 2^{\lambda - H + H'}$.

In our context, it states that the conditional min-entropy of the secret key after λ bits of leakage is at least H' but with probability $2^{\lambda - H + H'}$. We will use the lemma with $H' = 1$ because one bit of uncertainty is sufficient to apply Lemma 3.7 (witness indistinguishability) in Theorem 3.8.

Theorem 3.11 (Leakage Resilience) *Let $c_m = \omega(1)$ and let $L := \log(|D_s^m|) = mn \log(2d_s + 1)$ be the length of the secret key. The conditional min-entropy H_∞ of $\hat{\mathbf{s}}$, conditioned on $\mathbf{S} = h(\hat{\mathbf{s}})$ and a total secret-key leakage $f(\hat{\mathbf{s}})$ of $\lambda = \delta L = (1 - o(1))L$ bits, is positive with overwhelming probability.*

Proof. We prove a conservative lower bound on the amount of tolerated leakage because we treat the public key $\mathbf{S}$ as additional leakage. Therefore, we define a new total leakage

function $f'(\hat{\mathbf{s}}) = f(\hat{\mathbf{s}})||h(\hat{\mathbf{s}})$ with a total leakage of at most $\lambda' = \lambda + n\log(q)$ bits. Now, we apply Lemma 3.10 to f', λ', and H' with $\hat{\mathbf{s}}$ being the random variable. Observe that $H = L = mn\log(2d_s + 1)$. It yields

$$\text{Prob}\left[\, f'(\hat{\mathbf{s}}) \in Y \right] \geq 1 - 2^{\lambda + n\log(q) - L + 1}\,, \tag{4}$$

which we want to be overwhelming $\geq 1 - 2^{-p(n)}$. We take any function $p(n)$, $\omega(\log(n)) \leq p(n) \leq \mathcal{O}(n\log(n))$ and bound the relative leakage $\delta \leq 1 - \frac{p(n)+n\log(q)+1}{L} = 1 - \frac{\Theta(n\log(n))}{c_m\Theta(n\log(n))} = 1 - \frac{1}{\omega(1)} = 1 - o(1)$.

In consequence, (4) becomes $\geq 1 - 2^{\left(1 - \frac{p(n)+n\log(q)+1}{L}\right)L + n\log(q) - L + 1} = 1 - 2^{p(n)}$. Thus, $\delta L = (1 - o(1))L$ leakage bits yield a non-zero conditional min-entropy with probability $1 - 2^{-p(n)} \geq 1 - 2^{-\omega(\log(n))}$. □

4. Conclusions

We have shown how to construct an efficient and provably secure blind signature scheme based on the hardness of worst-case lattice problems. Our scheme has four moves, offers quasi-optimal performance, and it is leakage resilient in an almost optimal sense. Therefore, we expect our construction to withstand even subexponential-time and quantum computer attacks, as well as limited side-channel attacks against the secret key.

Acknowledgments

The author thanks Özgür Dagdelen, Marc Fischlin, Tibor Jager, Vadim Lyubashevsky, Chris Peikert, Michael Schneider, and Dominique Schröder for reviewing parts of this work and for very helpful and encouraging discussions. He also thanks the ananymous reviewers of ASIACRYPT 2010 for their valuable input.

References

[Abe01] Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 136–151. Springer, 2001.

[ADL+08] Y. Arbitman, G. Dogon, V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFTX: A proposal for the SHA-3 standard, 2008. In the First SHA-3 Candidate Conference.

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In Halevi [Hal09], pages 36–54.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, 2009.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108. ACM, 1996.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610. ACM, 2001.

[ANN06] Michel Abdalla, Chanathip Namprempre, and Gregory Neven. On the (im)possibility of blind message authentication codes. In David Pointcheval, editor, *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 262–279. Springer, 2006.

[AO09] Masayuki Abe and Miyako Ohkubo. A framework for universally composable non-committing blind signatures. In Matsui [Mat09], pages 435–450.

[BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the "one-more" computational problems. In Tal Malkin, editor, *CT-RSA*, volume 4964 of *Lecture Notes in Computer Science*, pages 71–87. Springer, 2008.

[BN06] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 390–399. ACM, 2006.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-rsa-inversion problems and the security of chaum's blind signature scheme. *J. Cryptology*, 16(3):185–215, 2003.

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In Yvo Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2003.

[BR93] Mihir Bellare and Pil Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS*. ACM, 1993.

[Cha82] David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.

[CKW04] Jan Camenisch, Maciej Koprowski, and Bogdan Warinschi. Efficient blind signatures without random oracles. In *Security in Communication Networks*, volume 3352 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 2004.

[CNS07] Jan Camenisch, Gregory Neven, and Abhi Shelat. Simulatable adaptive oblivious transfer. In Moni Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 573–590. Springer, 2007.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302. IEEE Computer Society, 2008.

[ECR10] ECRYPT2. Yearly report on algorithms and keysizes — report D.SPA.13, 2010. available at `http://www.ecrypt.eu.org/documents/D.SPA.13.pdf`.

[Fis06] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 60–77. Springer, 2006.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[FS09] Marc Fischlin and Dominique Schröder. Security of blind signatures under aborts. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 297–316. Springer, 2009.

[Gol04] Oded Goldreich. *The Foundations of Cryptography*, volume 1. Cambridge University Press, 2004.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *STOC*, pages 197–206. ACM, 2008.

[Hal09] Shai Halevi, editor. *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*. Springer, 2009.

[HKKL07] Carmit Hazay, Jonathan Katz, Chiu-Yuen Koo, and Yehuda Lindell. Concurrently-secure blind signatures without random oracles or setup assumptions. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 323–341. Springer, 2007.

[JLO97] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In Burton S. Jr. Kaliski, editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 1997.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer, 2008.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In Matsui [Mat09], pages 703–720.

[Len05] Arjen Lenstra. *The Handbook of Information Security*, chapter 114 — Key Lengths. Wiley, 2005. available at `http://www.keylength.com/biblio/Handbook_of_Information_Security_-_Keylength.pdf`.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2006.

[Lyu08a] Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In Ronald Cramer, editor, *Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2008.

[Lyu08b] Vadim Lyubashevsky. *Towards Practical Lattice-Based Cryptography*. PhD thesis, University of California, San Diego, 2008.

[Lyu09] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In Matsui [Mat09], pages 598–616.

[Mat09] Mitsuru Matsui, editor. *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*. Springer, 2009.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Halevi [Hal09], pages 18–35.

[Oka92] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer, 1992.

[Oka06] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 80–99. Springer, 2006.

[Poi98] David Pointcheval. Strengthened security for blind signatures. In Kaisa Nyberg, editor, *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, pages 391–405. Springer, 1998.

[PS97] David Pointcheval and Jacques Stern. New blind signatures equivalent to factorization (extended abstract). In *ACM Conference on Computer and Communications Security*, pages 92–99, 1997.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[RHOAGZ07] Francisco Rodríguez-Henríquez, Daniel Ortiz-Arroyo, and Claudia García-Zamora. Yet another improvement over the mu-varadharajan e-voting protocol. *Comput. Stand. Interfaces*, 29(4):471–480, 2007.

[RS10] Markus Rückert and Michael Schneider. Estimating the security of lattice-based cryptosystems. Cryptology ePrint Archive, Report 2010/137, 2010. `http://eprint.iacr.org/`.

[Rüc10] Markus Rückert. Adaptively secure identity-based identification from lattices without random oracles. In Juan A. Garay and Roberto De Prisco, editors, *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 345–362. Springer, 2010.

[Sch91] C.P. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4:161–174, 1991.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

A. Forking Lemma

The generalized Forking Lemma of Bellare and Neven [BN06] is a tool for proving security in the random oracle model. It provides a lower bound for the probability that a randomized algorithm outputs two related values when run twice with the same random tape but with a different random oracle.

Lemma A.1 (Lemma 1 in [BN06]) *Fix an integer $q \geq 1$ and a set H of size $h \geq 2$. Let A be a randomized algorithm that on input $x, h_1, \ldots, h_q$ returns a pair, the first element of which is an integer in the range $0, \ldots, q$ and the second element of which we refer to as a side output. Let IG be a randomized algorithm that we call the input generator. The accepting probability of A, denoted* acc*, is defined as the probability that $J \geq 1$ in the experiment $x \leftarrow_{\$} \mathsf{IG}; h_1, \ldots, h_q \leftarrow_{\$} H; (J, \sigma) \leftarrow_{\$} \mathsf{A}(x, h_1, \ldots, h_q)$. The forking algorithm F_{A} associated to A is the randomized algorithm that takes input x proceeds as follows:*

Algorithm $\mathsf{F}_{\mathsf{A}}(x)$
Pick coins ρ for A at random
$h_1, \ldots, h_q \leftarrow_{\$} H$

$(I, \sigma) \leftarrow \mathsf{A}(x, h_1, \ldots, h_q; \rho)$
If $I = 0$ *then return* $(0, \epsilon, \epsilon)$
$h'_I, \ldots, h'_q \leftarrow_{\$} H$
$(I', \sigma') \leftarrow \mathsf{A}(x, h_1, \ldots, h_{I-1}, h'_I, \ldots, h'_q; \rho)$
If $I = I'$ *and* $h_I \neq h'_{I'}$ *then return* $(1, \sigma, \sigma')$
Else return $(0, \epsilon, \epsilon)$.

Let $\mathsf{frk} = \mathrm{Prob}[\, b = 1 : x \leftarrow_{\$} \mathsf{IG}; (b, \sigma, \sigma') \leftarrow \mathsf{F}_{\mathsf{A}}(x)]$. *Then* $\mathsf{frk} \geq acc\left(\frac{acc}{q} - \frac{1}{h}\right)$.

B. Leakage-resilience for Blind Signatures

The following experiment $\mathsf{Exp}^{\mathsf{omf},\lambda\text{-}\mathsf{Leak}}_{\mathcal{U}^*,\mathsf{BS}}$ models leakage-resilience for unforgeability of blind signatures. The scheme $\mathcal{BS}$ is leakage-resilient with λ if there is no efficient adversary $\mathcal{U}^*$ for which the experiment outputs 1. The parameter λ is a function of the length L of the secret key and it controls the amount of tolerated leakage.

Experiment $\mathsf{Exp}^{\mathsf{omf},\lambda\text{-}\mathsf{Leak}}_{\mathcal{U}^*,\mathsf{BS}}(n)$
$\mathsf{H} \leftarrow_{\$} \mathcal{H}(1^n)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{BS.Kg}(1^n)$
$\{(M_1, \mathbf{s}_1), \ldots, (M_\jmath, \mathbf{s}_\jmath)\} \leftarrow_{\$} \mathcal{U}^{*\mathsf{H}(\cdot), \langle \mathcal{S}(\mathsf{sk}), \cdot \rangle^{\infty}, \mathsf{Leak}(\mathsf{sk}, \cdot)}(\mathsf{pk})$
Let ℓ be the number of successful interaction between $\mathcal{U}^*$ and the signer.
Let $f_1, \ldots, f_\kappa$ be the leakage queries of $\mathcal{U}^*$, each with output length λ_i.
Return 1 iff
1. $M_i \neq M_j$ for all $1 \leq i < j \leq \jmath$;
2. $\mathsf{BS.Vf}(\mathsf{pk}, \mathbf{s}_i, M_i) = 1$ for all $i = 1, \ldots, \jmath$;
3. $\ell + 1 = \jmath$;
4. $\sum_{i=1}^{\kappa} \lambda_i \leq \lambda(|\mathsf{sk}|)$.

C. Practical Parameters

Although worst-case guarantees are a good argument for lattice-based cryptography in general, we need to analyze the underlying average-case problem, the collision problem, directly before proposing concrete parameters. To this end, we use the results from [RS10], which provides a framework for choosing secure parameters for lattice-based cryptography.

All we need to apply the framework, is to feed it with valid parameter relations (cf. Table 2) and run the main reduction (one-more unforgeability). As a result, we end up with an instance of the collision problem $Col(\mathcal{H}(\mathbf{R}, m), D)$, or alternatively with the SIS (short integer solution) problem. Since [RS10] only deals with ℓ_2-norm we relax the adversaries task and assume it only needs to find a vector of Euclidean norm $\leq d_D\sqrt{mn}$. The resulting instance of SIS then yields a hardness estimate δ, which can be related to "bit security".

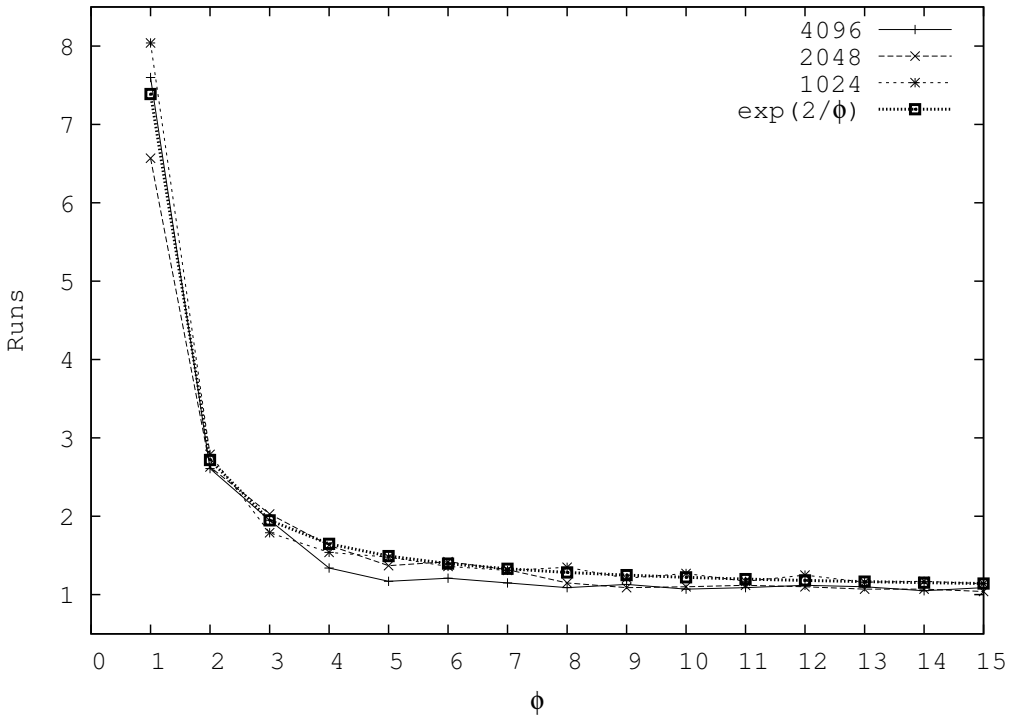

Figure 3: Average number of runs needed to complete in the signature protocol of BS for $\phi \in [1, 15] \cap \mathbb{Z}$

C.1. Optimization

Before we propose actual parameters, we show how to optimize the choice of parameters based on the constraints in Table 2. We discuss the parameters q, ψ, ϕ, and d_s.

Choosing q. From Table 2 and Theorem 2.2, we know that the worst-case to average-case reduction relies on having $q \geq 4mn\sqrt{n}\log(n)d_D = \widetilde{\Theta}(n^5\sqrt{n})$. For practical parameters, we typically arrive at $q \approx n^8$.

Choosing ψ. As noted before, we can safely set $\psi = 1$ and let the user handle an expected number of e restarts after Step 2 in the protocol. These restarts are performed locally and experiments show that they do not significantly affect the overall efficiency of the protocol.

Choosing ϕ. For ϕ, the situation is different because it controls the probability of a restart after Steps 3 and 4. Restarts at this point cannot be done locally and involve an increase in communication and computation costs. The influence becomes obvious when looking at Figure 3. It shows the number of required repetitions of the blind signature protocol for $n \in \{1024, 2048, 4096\}$ and $\phi \in [1, 15] \cap \mathbb{Z}$, averaged over 1000 random instances. As a comparison, it also shows the estimated number of repetitions $e^{2/\phi}$ from

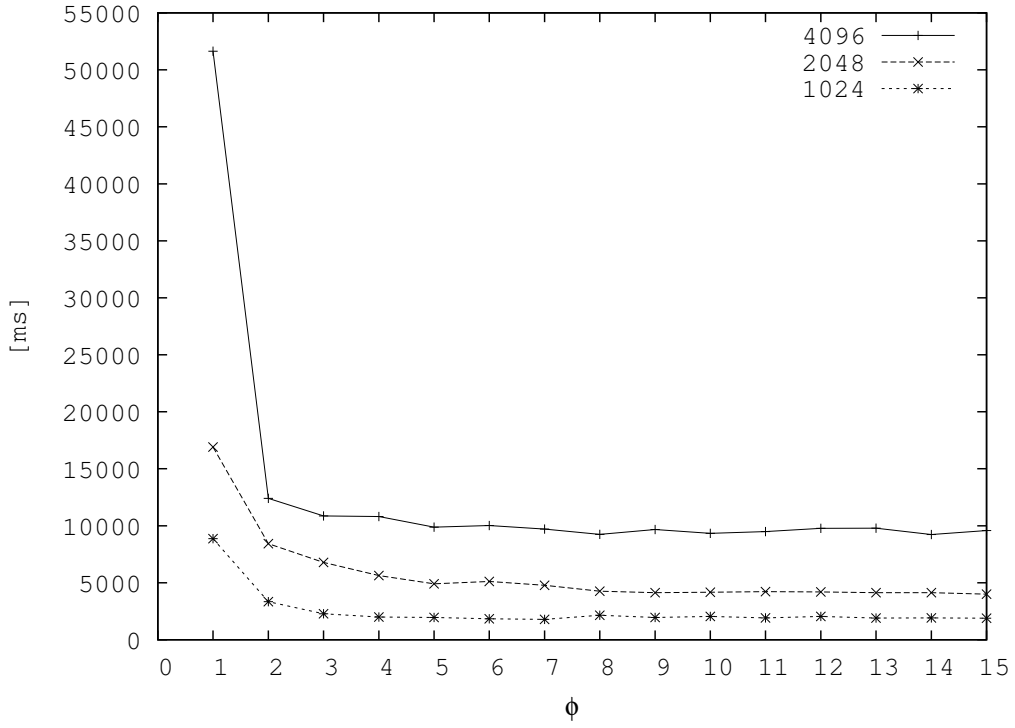

Figure 4: Average running time of signature protocol and the verification algorithm of BS for $\phi \in [1, 15] \cap \mathbb{Z}$ in milliseconds. The running times include eventual restarts.

Section 3. As expected, the behavior is almost independent of n. Our experiments show that one should at least choose $\phi \geq 4$ and by closely looking at the numbers it even makes sense to choose $\phi = 10$ in some cases. This observation is backed up by Figure 4, which shows the actual combined running time in milliseconds of signing and verification, averaged over 1000 random instances. Since our implementation is pretty straightforward without many optimizations, we believe that there is a lot of room for improvements.

When increasing ϕ, one has to keep in mind that a larger ϕ also increases d_D and requires a slightly stronger hardness assumption.

Choosing d_s. When looking at the constraint for m, basically $m > \log(q)/\log(2d_s)$, it is clear that a larger d_s allows us to choose a smaller m. This can dramatically decrease the required bandwidth and computational cost at the expense of a slightly larger secret key. We show the effect of a larger d_s in Figure 5 for $n = 1024$ and $\phi = 4$. Notice that we take the logarithm of d_s and round m to the next integer. So, in order to decrease m by a factor k, we need to choose d_s around 2^k. When removing the outliers, this leads to a perfect stair-stepped graph. Also, this increase comes at the price of having a larger d_D and requires stronger hardness assumption.

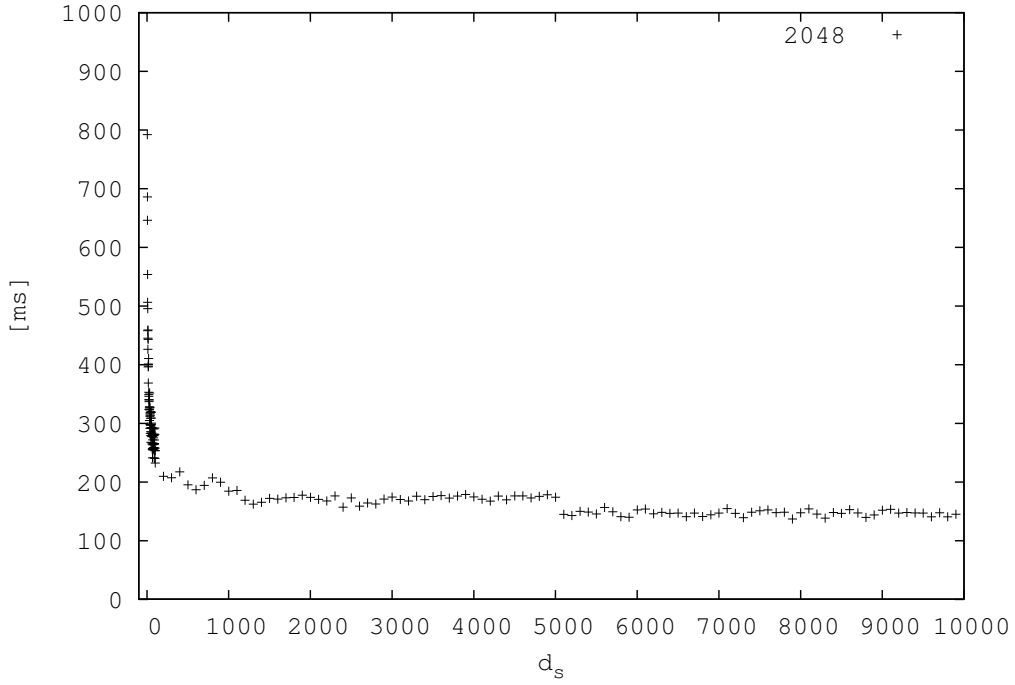

Figure 5: Average running time of signature protocol and the verification algorithm of BS for $d_s \in [1, 1024] \cap \mathbb{Z}$ and $n = 1024$ and $\phi = 4$.

C.2. Secure Parameters

Table 3 shows a few exemplary parameter sets for current (76 bit) and medium (102 bit) security levels. According to `www.keylength.com`, they correspond to security until the years 2012 and, 2050, respectively. We leave out the parameter sets for future (256 bit / year 2282) security.

For both security levels, we propose three parameter sets. The first requires the mildest hardness assumption and uses the smallest modulus. The second minimizes the number of repetitions and the third is simultaneously optimized for computational cost and bandwidth including restarts. The optimization goal is denoted in bold face. Depending on the application scenario, other trade-offs are possible.

D. Supporting Lemmas

D.1. Completeness

Proving Lemmas 3.1 and 3.2 concludes the discussion of completeness in Theorem 3.3. Lemma 3.1 shows that the number of aborts/restarts in the protocol is small. Lemma 3.2 ensures that multiplying two polynomials in $\mathbf{R}$ only slightly increases the norm of the resulting coefficient vector.

Parameter	Current I	Current II	Current III	Mid I	Mid II	Mid III
n	1024	1024	1024	2048	2048	2048
q	$\approx 2^{78}$	$\approx 2^{85}$	$\approx 2^{81}$	$\approx 2^{85}$	$\approx 2^{91}$	$\approx 2^{94}$
ϕ	1	8	4	1	10	4
d_s	1	1	283	1	1	241080
m	79	86	9	85	92	5
Repetitions	7.13	**1.32**	1.55	7.67	**1.16**	1.23
Secret key	15.7 kB	17 kB	10.3 kB	33.7 kB	36.5 kB	23.6kB
Public key	9.8 kB	10.6 kB	10,2 kB	21.2 kB	22.9 kB	23.6 kB
Signature	529.4 kB	643.4 kB	**66.9 kB**	1228.6 kB	1487.9 kB	**89.4 kB**
Communication	2706.4 kB	589.5 kB	**95.2 kB**	6771.5 kB	1199.8 kB	**119.1 kB**
KeyGen	296 ms	369 ms	**37 ms**	674 ms	843 ms	**52 ms**
Signing	7629 ms	1819 ms	**220 ms**	19000 ms	3656 ms	**283 ms**
Verification	226 ms	293 ms	**33 ms**	1620 ms	679 ms	**57 ms**

Table 3: Exemplary parameters for the blind signature scheme in Section 3. Sizes in kilo bytes (kB) and times in milliseconds (ms) are rounded to the nearest integer.

Lemma 3.1 Let $k = \Omega(n)$, $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^k$ with *arbitrary* $\mathbf{a} \in \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \leq A\}$ and *random* $\mathbf{b} \leftarrow_\$ \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \leq B\}$. Given $B \geq \phi k A$ for $\phi \in \mathbb{N}_{>0}$, we have $\operatorname{Prob}_{\mathbf{b}}[\,\|\mathbf{a}-\mathbf{b}\|_\infty \leq B - A] > \frac{1}{e^{1/\phi}} - o(1)$.

Proof (Lemma 3.1). Observe that $\operatorname{Prob}[\,\|\mathbf{a}-\mathbf{b}\|_\infty \leq B-A] = \operatorname{Prob}[\,|a_i - b_i| \leq B - A]^k$ and that the b_i need to be in the range $[-(B-A)+a_i, B-A+a_i] \subseteq [-B, B]$ for that. Therefore, the probability is

$$\left(\frac{2(B-A)+1}{2B+1}\right)^k > \left(1 - \frac{A}{B}\right)^k \geq \left(1 - \frac{1/\phi}{k}\right)^k.$$

By the series expansion at infinity, this is at least $\frac{1}{e^{1/\phi}} - o(1)$ for an $o(1)$ term that vanishes like $1/k$. □

Lemma 3.2 For any two polynomials $\mathbf{a}, \mathbf{b} \in \mathbf{R}$, we have $\|\mathbf{ab} \bmod (X^n+1)\|_\infty \leq n \|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty$.

Proof (Lemma 3.2). Note that $\mathbf{c} = \mathbf{ab} \bmod (X^n+1) = \sum_{i=0}^{n-1} a_0 \mathbf{b} X^i \bmod (X^n+1)$. Hence, we have $\|\mathbf{c}\|_\infty = \left\|\sum_{i=0}^{n-1} a_0 \mathbf{b} X^i \bmod (X^n+1)\right\|_\infty \leq n \|\mathbf{a}\|_\infty \max_{i=0,\ldots,n-1}\{\mathbf{b}X^i \bmod (X^n+1)\}$, where the last term is reminiscent of an operator norm. For our particular ring polynomial X^n+1, this norm is easy to evaluate because $\mathbf{b}X = b_0 X + b_1 X^2 + \cdots + b_{n-2}X^{n-1} + b_{n-1}X^n \bmod (X^n+1) = -b_{n-1} + b_0 X + b_1 X^2 + \cdots + b_{n-2}X^{n-1}$. Therefore, we have $\left\|\mathbf{b}X^i \bmod (X^n+1)\right\|_\infty = \|\mathbf{b}\|_\infty$ and $\|\mathbf{c}\|_\infty \leq n \|\mathbf{a}\|_\infty \|\mathbf{b}\|_\infty$. □

D.2. Blindness

Lemma 3.4 establishes blindness in Theorem 3.5. It guarantees that the every output by the user is distributed independently of the signed message via a careful choice of parameters and sets.

Lemma 3.4 Let $k \in \mathbb{N}$, $\mathbf{a}, \mathbf{a}', \mathbf{b} \in \mathbb{Z}^k$ with *arbitrary* $\mathbf{a}, \mathbf{a}' \in \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \leq A\}$, a *random* $\mathbf{b} \leftarrow_\$ \{\mathbf{v} \in \mathbb{Z}^k : \|\mathbf{v}\|_\infty \leq B\}$ for $B > A$. We define the random variables $\mathbf{c} \leftarrow \mathbf{a} - \mathbf{b}$ and $\mathbf{c}' \leftarrow \mathbf{a}' - \mathbf{b}$ if $\max\{\|\mathbf{a} - \mathbf{b}\|_\infty, \|\mathbf{a}' - \mathbf{b}\|_\infty\} \leq B - A$, otherwise, we resample $\mathbf{b}$. Then, $\Delta(\mathbf{c}, \mathbf{c}') = 0$.

Proof (Lemma 3.4). By definition, the statistical distance is
$\frac{1}{2} \sum_{\mathbf{c}:\|\mathbf{c}\|_\infty \leq B-A} \left| \Prob_{\mathbf{b}}[\mathbf{a} - \mathbf{b} = \mathbf{c}] - \Prob_{\mathbf{b}}[\mathbf{a}' - \mathbf{b} = \mathbf{c}] \right| = \frac{1}{2} \sum_{\mathbf{c}} \left| \Prob_{\mathbf{b}}[\mathbf{b} = \mathbf{a} - \mathbf{c}] - \Prob_{\mathbf{b}}[\mathbf{b} = \mathbf{a}' - \mathbf{c}] \right|$.
Observe that $\max\{\|\mathbf{a} - \mathbf{c}\|_\infty, \|\mathbf{a}' - \mathbf{c}\|_\infty\} \leq A + (B - A) = B$ and $\|\mathbf{b}\|_\infty \leq B$. Hence, the probability in either case is $1/(2B+1)^k$ and the statistical distance is 0. □

D.3. One-more Unforgeability

Lemma 3.6 ensures that all secret keys collide with at least one alternative secret key under h. In combination with Lemma 3.7, which proves witness indistinguishability of the signature issue protocol, this allows us to build a reduction algorithm in Theorem 3.8 that correctly simulates the signer and breaks $Col(\mathcal{H}(\mathbf{R}, m), D)$ with the help of a forger.

Lemma 3.6 Let $h \in \mathcal{H}(\mathbf{R}, m)$. For every secret key $\hat{\mathbf{s}} \leftarrow_\$ D_s^m$, there is a second $\hat{\mathbf{s}}' \in D_s^m \setminus \{\hat{\mathbf{s}}\}$ with $h(\hat{\mathbf{s}}) = h(\hat{\mathbf{s}}')$ (with overwhelming probability).

Proof (Lemma 3.6). All functions in the family $\mathcal{H}(\mathbf{R}, m)$ are compressing when applied to the domain D_s^m for our choice of parameters because $|D_s^m| = (2d_s + 1)^{mn} > 3^{mn} > (2d_s)^{n \log(q)/\log(2d_s)} > 2^{n \log(q)} = q^n = |\mathbf{R}|$. Therefore, all but at most q^n elements in D_s^m do not collide. The probability of selecting such an element is at most $(q/(2d_s+1)^m)^n < 2^{-n \log(q) \log(2d_s+1)/\log(2d_s)} < 2^{-n \log(q)} = 2^{-\Omega(n \log(n))}$. □

Lemma 3.7 Let $h \in \mathcal{H}(\mathbf{R}, m)$ and $\mathbf{S} \in \mathbf{R}$. For any message M and any two secret keys $\hat{\mathbf{s}}, \hat{\mathbf{s}}' \in D_s^m$ with $h(\hat{\mathbf{s}}) = \mathbf{S} = h(\hat{\mathbf{s}}')$, the resulting protocol views $(\mathbf{Y}, \epsilon^*, \hat{\mathbf{z}}^*)$ and $(\mathbf{Y}', \epsilon^{*\prime}, \hat{\mathbf{z}}^{*\prime})$ are indistinguishable.

Proof (Lemma 3.7). The argument is an adaptation of [Lyu08b, Theorem 5.5]. We interpret the components of the view as random variables. Firstly, observe that $\mathbf{Y}$ and $\mathbf{Y}'$ are chosen independently of the secret key. Secondly, ϵ^* and $\epsilon^{*\prime}$ are independent of a particular $\hat{\mathbf{y}} \in h^{-1}(\mathbf{Y}) \cap D_y^m$ because $\mathbf{Y}$ statistically hides $\hat{\mathbf{y}}$. Finally, we need to argue about the indistinguishability of $\hat{\mathbf{z}}^*$ and $\hat{\mathbf{z}}^{*\prime}$. Let ϵ^* be any challenge and let $\hat{\mathbf{z}}^* = \hat{\mathbf{s}}\epsilon^* + \hat{\mathbf{y}} \in G_*^m$. Then, we can set $\hat{\mathbf{y}}' \leftarrow \hat{\mathbf{y}} + \hat{\mathbf{s}}\epsilon^* - \hat{\mathbf{s}}'\epsilon^*$ for which $\hat{\mathbf{z}}^* = \hat{\mathbf{s}}'\epsilon^* + \hat{\mathbf{y}}'$. We need to show that $\hat{\mathbf{y}}' \in h^{-1}(\mathbf{Y}) \cap D_y^m$. This implies that for every $\hat{\mathbf{y}}$ (for $\hat{\mathbf{s}}$), there

is a $\hat{\mathbf{y}}'$ (for $\hat{\mathbf{s}}'$) that yields the same output. Thus, the probability of a restart would also be equal. Clearly, $\hat{\mathbf{y}}' \in h^{-1}(\mathbf{Y})$ because $h(\hat{\mathbf{y}}') = \mathbf{Y} + \mathbf{S}\epsilon^* - \mathbf{S}\epsilon^* = \mathbf{Y}$. Furthermore, $\|\hat{\mathbf{y}}'\|_\infty \leq \|\hat{\mathbf{z}}^*\|_\infty + \|\hat{\mathbf{s}}'\epsilon^*\|_\infty \leq d_y - nd_s d_{\epsilon^*} + nd_s d_{\epsilon^*} = d_y$ by Lemma 3.2 and we can conclude $\hat{\mathbf{y}}' \in D_y^m$. □